

# Combining Contexts and Ontologies: A Case Study

Mariela Rico<sup>1</sup>, Ma. Laura Caliusco<sup>2</sup>, Omar Chiotti<sup>3</sup>, and Ma. Rosa Galli<sup>3</sup>

1 CIDISI-UTN-FRSF, Lavaysse 610, 3000, Santa Fe, Argentina,  
mrico@frsf.utn.edu.ar

2 CIDISI-UTN-FRSF, CONICET, Lavaysse 610, 3000, Santa Fe, Argentina,  
mcaliusc@frsf.utn.edu.ar

3 INGAR-CONICET-UTN, Avellaneda 3657, 3000, Santa Fe, Argentina  
{chiotti,mrgalli}@ceride.gov.ar

**Abstract.** In the last years different proposals that integrate ontologies and contexts, taking advantages from their abilities for achieving information semantic interoperability, have arisen. Each of them has considered the problematic from different perspectives. Particularly, through an actual case study this paper shows which are the problems related to support information semantic interoperability between different parties involved in a collaborative business relation over the Internet. Furthermore, this paper analyzes those problems from previous proposals perspective and highlights the weaknesses and strengths of them.

## 1 Introduction

Due to the necessity of creating explicit models of the semantic associated to information with different objectives, such as knowledge management, information interoperability and so on; two different approaches have separately arisen: ontologies and contexts. On the one hand, ontologies represent an explicit specification of conceptualization shared within a domain [1]. On the other hand, a context could be defined in a general way as an environment within a fact is considered. This environment is characterized by a set of axioms and true facts [2].

Ontologies and contexts have both strengths and weaknesses. On the one hand, shared ontologies define a common understanding of specific terms, and thus make it possible interoperable systems on a semantic level. On the weak side, ontologies can be used only as long as consensus about their content is reached. On the other hand, contexts encode not shared interpretation schemas of individuals and they are easy to define and to maintain because they can be constructed with a limited consensus with the other parties. On the weak side, since contexts are local to parties,

communication can be achieved only by constructing explicit mappings among the elements of the contexts of the involved parties [3].

Therefore, considering that the strengths of ontologies are the weaknesses of contexts and vice versa, and therefore that they can be seen as complementary approaches, a new research area arises which integrate both concepts to achieve semantic interoperability of information. The most relevant works that integrate contexts and ontologies have been developed in the following areas: Semantic Web [3], information system interoperability [4, 5], document classification [6, 7], and context-aware applications [8, 9]. These works consider the problem from different perspectives and we think that some of these perspectives have to be integrated into one in order to efficiently describe semantics to achieve information interoperability.

The aim of this paper is to analyze how ontologies and contexts could be related in order to support semantic interoperability of information interchanged between parties involved in a collaborative business relation. To achieve this objective, first, we present a case study in which through examples we analyzed different ways of ontologies and contexts combination. Then, some related works are analyzed and compared with the results of the case study. Finally, our conclusions and future works are presented.

## 2 Case Study: Partner–To–Partner Collaborative Model

The integration problem of a supply chain via Internet is a typical problem of information interoperability, where the business documents must be interpreted by the information systems of different trading partners. This involves the collaboration in one or more stages of the business process, from production planning to sale stage.

A collaborative production planning model defines a collaborative business process divided into three subprocesses; each of them implies a different collaborative business process [10]. These subprocesses are:

- Consensus at level of Production Aggregated Planning (PAP), which objective is that enterprises reach a consensus about a material provision plan at PAP level. Here, trading partners agree about which products they will collaborate (at the product family level), in which periods of time (horizon is between 6 and 18 months) and approximate quantities of them.
- Consensus at level of Production Master Program (PMP), with which enterprises have to arrive to a consensus about a material provision scheduling at PMP level. At this level, products are defined at highest detail level required by the manufacturer enterprise. Also, periods of time and quantities are specified at a higher level of detail. The customer specifies required material quantity and due date. The supplier defines the date and the size of provision orders, considering what has been agreed in the frame agreement. This is “probable” information.
- Consensus at level of Provision Orders Program (POP), which objective is that trading partners reach a consensus about the definition of a provision orders scheduling. At this level, “sure” information is handled. Periods of time 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.

A collaborative relation management implies coordinating: private processes that are executed by an enterprise; and collaborative processes that are jointly executed

by trading partners. The latest are defined as abstract ones; and to implement it, each trading partner has to define a business interface process (IP). This IP is responsible for the role that a partner plays in a collaborative process, and the invocation and execution of those private processes required for carrying it out. Furthermore, to interchange information Electronic Business Documents (EBDs) are sent as a part of messages exchanged between trading partners. EBDs are standardized data structures that replace traditional business documents; they support the exchange of information required to execute collaborative processes, like order forecasting [2].

Fig. 1 shows a collaborative relation between a brewery (supplier) and one of its clients (retailer). When the IP receives an EBD, it has to translate the information contained in the document to the private processes according to the semantic of corresponding enterprise domains. Then, when the IP has to send an EBD, it has to populate the EBD with the information generated by the corresponding enterprise domains according to the collaborative process semantics. So, to achieve information system interoperability, the IP has to solve a number of conflictive situations at the semantic level that are analyzed in the rest of the paper.

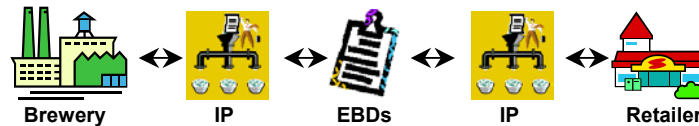


Fig. 1. A collaborative relation between a brewery (supplier) and one of its clients

Supposing a collaborative relation like that above showed, some of the EBDs used in each collaborative subprocess could have the following structures:

Table 1. EBD’s structure to interchange in the PAP subprocess

Horizon: <i>[months]</i>		
Period	Product	Quantity
<i>[months]</i>	<i>[Local, Strong, NKH, DUF]</i>	<i>[liters]</i>

Table 2. EBD’s structure to interchange in the PMP subprocess

Horizon <i>[weeks]</i>										
Period	Product				Quantity	Price		SuggestedPrice		Payment Way
	Type	Trademark	Packaging			Coinage	Amount	Coinage	Amount	
<i>[week]</i>	<i>[Local, Strong, NKH, DUF]</i>	<i>[EFS, HCS, NKH, DUF]</i>	<i>[cm³]</i>	<i>[can, bottle]</i>	<i>[units of pack]</i>	<i>[\$, US\$]</i>	<i>[float]</i>	<i>[\$, US\$]</i>	<i>[float]</i>	<i>[string]</i>

In the PMP, for each stage, the quantities are defined in units of pack and new information is incorporated such as price, suggested price and payment way. Generally, time horizon comprises two months, i.e. eight weeks.

**Table 3.** EBD's structure to interchange in the POP subprocess

Date	Product				Quantity	Price		SuggestedPrice		Payment Way
	Type	Trade mark	Packaging Size	Type		Coinage	Amount	Coinage	Amount	
[day]	[Local, Strong, NKH, DUF]	[EFS, HCS, NKH, DUF]	[cm <sup>3</sup> ]	[can, bottle]	[units of pack]	[\$, US\$]	[float]	[\$, US\$]	[float]	[string]

In Table 3, Date column indicates an accurate day in which the products will be available to be dispatched. In a POP, are detailed the quantities, products, prices and payment ways that must be dispatched in the indicated date.

When two enterprises establish a collaborative relation, they have to solve semantic interoperability problems [11]. For that, trading partners must define a way to represent the interchanged information semantic. From this perspective, there are two possible scenes: to develop a single ontology which describes all the EBDs, or to develop an ontology for each EBD. Before analyzing these scenes, it is necessary to clarify some ideas: independently from both scenes, the ontology used to represent the EBDs information semantics is not a global or general one, but only an ontology that describes documents semantic, which was agreed by both partners as part of the collaborative process. At the same time, these EBDs will be processed by partners' private processes; and these may affect different enterprise departments, which usually belong to different domains. That is, at the same enterprise, the same afore mentioned problems of semantics interoperability exist. Then, it is clear that each enterprise has its own ontologies to describe the semantics of its systems and internal areas. These ontologies are different from those agreed with the trading partners.

In spite of afore mentioned, we will concentrate now in the problematic of achieving semantic interoperability at level of EBDs. Following, each of those scenes is analyzed considering possibilities of integrating contexts and ontologies.

## 2.1 Use of the Same Ontology to Describe the Interchanged Information

Observing the tables that represent the needed information, we can see that there are repeated data in all of them although with particular features. Therefore, it is possible to suppose the use of a single EBD that must be defined as at syntactic as at semantic level to achieve the information interoperability at content level [4]. That is, it is necessary to describe how the information is structured and the meaning of that.

Fig. 2 presents an ontology schematic representation which describes "part of" and "is a" relations between terms of the EBD's structure. This structure had been defined in a consensual way to be used in each stage of the process. In this figure it is considered that the value by default of the relation cardinality is 1. According to this figure, a Plan can be compound by a Horizon or none. By other side, a Plan is compound by one to n Products. Each Product has associated a Quantity to be solicited or supplied, and a Period in which the Product will be solicited or delivered. Finally, a Plan can indicate a Price for each Product, a sale Suggested Price of it, and the Payment Way. Furthermore, in Fig. 2 it is represented that Production Planning, Production Program, and Provision Program are a Plan.

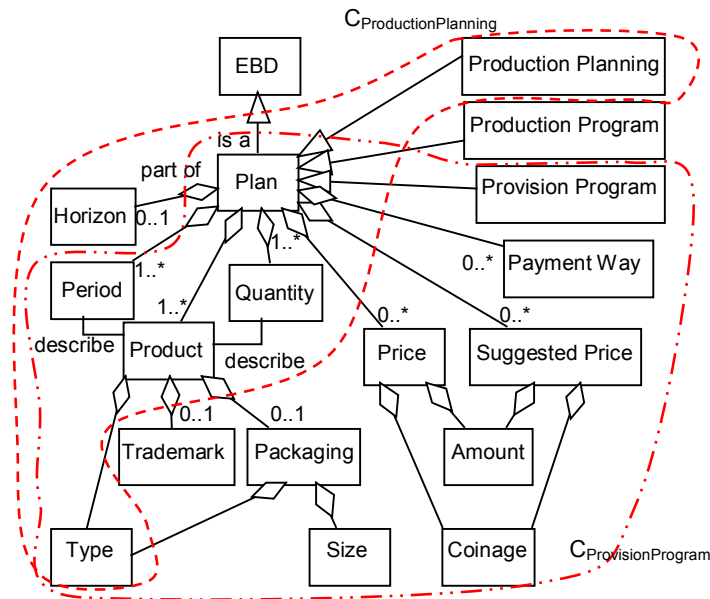


Fig. 2. Ontology of the consensual EBD

The scheme of Fig. 2 is the base to define a shared ontology between trading partners involved in a collaborative relation. That is, according to different ontology development methodologies [12, 13] one of the design stages is the conceptualization stage. This stage can be divided into two tasks that facilitate the definition of ontology: the task to represent the concepts by means of terms and its relations, and the axiomatization task (to add semantic restrictions and relations to the ontology). In this paper are only analyzed examples about the first task.

If the *Type* term of Fig. 2 is considered, this is associated to the *Packaging* and *Product* terms. Even though *Type* has the same semantics, since it describes the class or nature of the concepts which it is associated to, the possible values it may take are different. In the case of *Packaging*, *Type* can be *Can* or *Bottle*. On the other hand, for *Product*, *Type* can be *Local*, *Strong*, *DUF* or *NKH*. This presents an ambiguity problem that could be resolved replacing the term *Type* by *PackagingType* and *ProductType*. In this way, however, terms are unnecessarily added to the ontology, and this practice could lead to a size increase. A better solution would be to consider *Product* and *Packaging* as different contexts inside of which the term *Type* is interpreted (Fig. 3). In this example, the contexts are been generated with the purpose of resolving name ambiguities [14]. This is solved in other technologies with the namespace concept, which is a space (context) in which object names are unique.

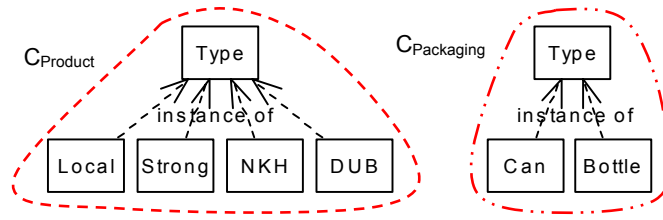


Fig. 3. Contexts that represent name spaces

Now, if it is analyzed the *Plan* term in Fig. 2, this presents different structures depending on if it represents the information of a *Production Planning* in the PAP subprocess, a *Production Program* in the PMP subprocess, or a *Provision Program* in the POP subprocess. Hence, in this case a concept changes its structure according to for what subprocess it is used. In this way, it can be considered that the ontology presented in Fig. 2 includes three contexts which were originated as a consequence of the existence of different situations [2]. According to the context, different terms of the EBD ontology must be considered to the document semantic description. In Fig. 2 are only shown two contexts not to interfere with its comprehension.

Then, as a conclusion of the discussed scene in this section, it can be said that an ontology can describe many contexts, and that a context can contain other contexts.

### 2.2 Use of Different Ontologies to Describe the Interchanged Information

In this scene it is supposed that different EBD’s structures are used in the same collaborative subprocess. In this way, it would have several smaller ontologies to manage in each subprocess.

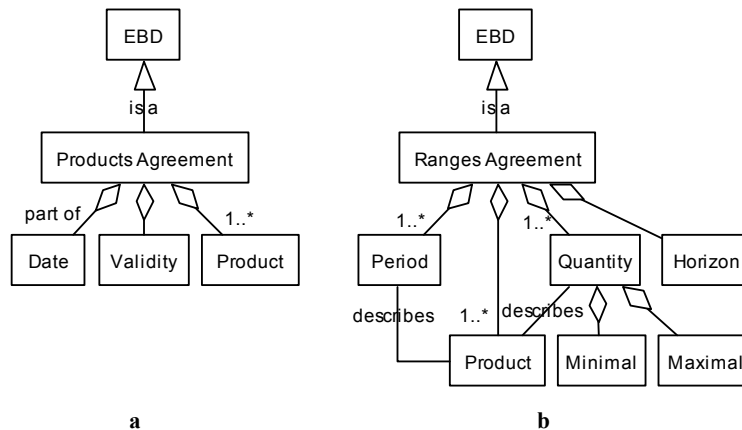
If the Production Aggregated Planning subprocess is considered as an example, this is divided into many stages [10]: Collaborative agreement, Aggregated planning, Acceptance / counterproposal, and Definitive acceptance.

In the Collaborative agreement stage, trading partners must agree on which products they will go to collaborate and, if they will go to work upon ranges of values, which are the ranges of quantities that the supplier partner is involved to deliver to the client partner. Table 4 shows the EBDs structures to be used to interchange this information.

Table 4. EBDs structures with a collaborative products list and the quantities ranges to supply

Date of agreement: <i>[date]</i>	Horizon: <i>[months]</i>		
Validity of agreement: <i>[months]</i>	Period	Product	Quantity
Product			Minimal
<i>[Local, Strong, NKH, DUF]</i>	<i>[months]</i>	<i>[Local, Stronge, NKH, DUF]</i>	<i>[liters] [liters]</i>

Fig. 4 shows the schematic representation of ontologies shared by the collaborative trading partners. Fig. 4a shows the ontology associated to the collaborative products list in so far as Fig. 4b shows the ontology associated to the ranges of quantities to supply.



**Fig. 4.** Shared ontologies in the Collaborative agreement stage of PAP subprocess. a) Products agreement ontology b) Ranges agreement ontology

As well as in the previous scene each subprocess was treated as a different context, also we can think that in each subprocess each stage constitutes a context. That is, the PAP context could be formed by contexts: Collaborative agreement, Aggregated Planning, Acceptance / Counterproposal, and Definitive acceptance.

If it is considered the Collaborative agreement context, this is compound by two ontologies that describe the corresponding EBDs. Analyzing the terms of these ontologies, it is observed that *Product* appears in both of them with the same possible values (*Local*, *Strong*, *NKH*, and *DUF*) but different relations. Though, this not represents a problem since the term belongs to two different ontologies.

Up to here then, it is not evidenced the requirement to make explicit the context. Each partner of collaborative process would be capable to process both types of EBDs without ambiguities. Though, as well as the Production Aggregated Planning subprocess is divided in stages, also the other two subprocesses are [10]. In particular, the Production Master Program subprocess also has a stage named Collaborative agreement. Therefore, it could be used the same EBDs of the Collaborative agreement in the PAP subprocess. Table 5 shows the EBDs structures to be used to interchange information of this stage.

If Tables 4 and 5 are compared, it is observed that the difference between them locates in the possible values of *Product* column. *Product*, in the Collaborative agreement context of PAP, can takes the *Local*, *Strong*, *NKH*, or *DUF* values, whereas that in the Collaborative agreement context of PMP can corresponds to *EFS*, *HCS*, *NKH*, or *DUF*. That is, even though the terms of the structures are the same, the allowed values vary because those represent different concepts. In the Collaborative agreement context of PAP, the *Product* term represents the concept of aggregated product. In so far as in the Collaborative agreement context of PMP, the *Product* term represents the concept of disaggregated product. Hence, to be possible to the partner of the collaborative process to treat correctly the received EBD, he needs to know of which of the subprocess the EBD correspond, i.e., it is necessary to make explicit the context.

**Table 5.** EBDs structures with a collaborative products list and the quantities ranges to supply for the PMP subprocess

Date of agreement: <i>[date]</i>	Horizon: <i>[months]</i>		
Validity of agreement: <i>[months]</i>	Period	Product	Quantity
Product			Minimal
<i>[EFS, HCS, NKH, DUF]</i>	<i>[months]</i>	<i>[EFS, HCS, NKH, DUF]</i>	<i>[liters] [liters]</i>

Summarizing, in this scene had been defined one ontology to each EBD, considered as different contexts. This avoids ambiguity problems. Though, when the same EBDs are reused in different situations (different situational contexts) to that which originate them, ambiguity problems can appear and hence the requirement to make explicit the situational context.

### 3 Related Works

The objective of this section is to show how the ideas, presented in previous works in different knowledge disciplines, about contexts and ontologies combination are related with the results of the case study presented in the previous section.

The most relevant works that integrate contexts and ontologies have been developed during the last years in the following areas: Semantic Web [3], system information interoperability [4, 5], document classification [6, 7], and context-aware applications [8, 9].

The idea of using contexts to disambiguate terms (synonymous and homonymous) presented in Section 2.1, has been discussed by DeLeenheer and Moor [7]. However in this case, the authors state that the context represents the information source from which terms are extracted and they do not consider another term as a possible context generator. We have shown in the case study that a term can be a context generator as the case of Product and Packaging terms.

Furthermore, from Fig. 2 we can deduce that ontology concepts are the union of context sets. This idea is similar to that presented by Segev and Gal [6] in the document classification area, who proposes a formal mechanism with the objective of mapping contexts, which are views of a domain automatically extracted, to ontologies concepts. This mechanism is used by the authors for routing mails (from the context are extracted) within a government organization composed by departments each of them described by an ontology.

In addition, it is possible to observe from Fig. 2, that there is an intersection between contexts and an inclusion of contexts. Furthermore, context could change along time. This idea is similar to that considered by Gu et al. [9] in the context-aware applications, who present a context model based on an ontology in order to support diverse tasks within intelligent environments. This model supports the representation of semantic contexts by defining a high level ontology that capture the general context knowledge about the physical world within pervasive computing environments, and providing a set of domain specific ontologies that define the details of general concepts and their properties within each sub domain (context into other context).



As a conclusion from two previous paragraphs, the case study shows that it is possible to have union, intersection and inclusion of contexts, but none of the analyzed proposals manage all the possibilities.

In contrast, to achieve the semantic interoperability among heterogeneous information systems, Wache and Stuckenschmidt [5] propose the use of a shared terminology of properties used to define different concepts. This terminology has to be enough general to be used through different information sources (contexts) to be integrated but enough specific to make possible significant definitions. Then, the authors use this basic shared vocabulary to derived relations into a shared ontology.

Finally, Bouquet et al. [3] present the idea of contextual ontologies. The authors say that an ontology is contextualized when its contents are kept local, and therefore not shared with other ontologies, and mapped with the contents of other ontologies via explicit context mappings. This idea of contextual ontologies does not fit into the problematic of semantic interoperability between trading partners we are putting forward because we have seen in the case study that both concepts are related in other ways. However, in [3] is discussed the mapping between concepts, which is not considered in this paper.

#### 4 Conclusions and Future Works

In this paper, through an actual case study, we have analyzed the problematic of integrating contexts and ontologies in order to support the interoperability of a collaborative business process over the Internet. We have presented two situations: one in which an ontology involves different contexts and another in which one context involves different ontologies. In both cases, we can observe that the document reutilization for information interchange drives the necessity to make the context explicit. Furthermore, we have seen that none of previous proposals solve all interoperability problems of a collaborative business process.

From the case study presented in Section 2 we can enumerate some advantages from contexts and ontologies combination:

- *To reduce the ontology size.* Working with a universe of reduced terms makes the tasks of information retrieval and interoperability more easy and efficient.
- *To show important details.* The context use allows the creation of concept groups which are important for a particular situation, hiding those concepts that are not relevant.
- *Information sources reutilization.* It is possible to have a unique data structure (tables or XML documents) associated to different contexts, in which the information within these structures be processed according to the context features where the information has to be integrated.

These advantages are not the only ones. Future research will focus on defining other case studies in order to discover other advantages. However, these advantages are significant enough to think about integrating ontologies and context to represent information semantic.

Furthermore, it is necessary to analyze the effort required to achieve the combination of contexts and ontologies. Also, we will study the problematic of

mapping between private ontologies of trading partners and EBDs ontologies, considering involved contexts and their representations.

## References

1. T.R. Grüber, Toward Principles for the Design of Ontologies Used for Knowledge Sharing, *International Journal of Human-Computer Studies*, Special Issue on Formal Ontology in Conceptual Analysis and Knowledge Representation, (1993).
2. M. L. Caliusco, A Semantic Definition Support of Electronic Business Documents in e-Collaboration (Universidad Tecnológica Nacional, F.R.S.F., 2005). ISBN 987-43-9158-8.
3. P. Bouquet, F. Giunchiglia, F. van Hamelen, L. Serafini, and H. Stuckenschmidt, Contextualizing Ontologies, *Journal of Web Semantics*, 1(4), 325-343 (2004).
4. A.P. Sheth, in: *Interoperating Geographic Information Systems/ Changing Focus on Interoperability in Information Systems: from System, Syntax, Structure to Semantics*, edited by M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman (Kluwer Academic Publishers, Norwell, Massachusetts, USA, 1999), pp. 5-30.
5. H. Wache, and H. Stuckenschmidt, in: *Modeling and Using Context: Third International and Interdisciplinary Conference/ Practical Context Transformation for Information System Interoperability*, edited by V. Akman, P. Bouquet, R. Thomason, and R.A. Young (Springer-Verlag GmbH, Dundee, UK, 2001), pp. 367-380.
6. A. Segev, and A. Gal, in: *Contexts and Ontologies: Theory, Practice and Applications: Papers from the 2005 AAI Workshop/ Putting Things in Context: a Topological Approach to Mapping Contexts and Ontologies*, edited by P. Shvaiko, J. Euzenat, A. Leger, D.L. McGuinness, and H. Wache (AAAI Press, Menlo Park, California, 2005), pp. 9-16.
7. P. De Leenheer, and A. de Moor, in: *Contexts and Ontologies: Theory, Practice and Applications: Papers from the 2005 AAI Workshop/ Context-driven Disambiguation in Ontology Elicitation*, edited by P. Shvaiko, J. Euzenat, A. Leger, D.L. McGuinness, and H. Wache (AAAI Press, Menlo Park, California, 2005), pp. 17-24.
8. H. Chen, T. Finin, and A. Joshi, An Ontology for Context-Aware Pervasive Computing Environments, *The Knowledge Engineering Review*, Special Issue on Ontologies for Distributed Systems, 18(03), 197-207, 2003.
9. T. Gu, X. Wang, H. Pung, and D. Zhang, in: *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference/ An Ontology-based Context Model in Intelligent Environments* (San Diego, California, USA, 2004).
10. P. Villarreal, M.L. Caliusco, M.R. Galli, E. Salomone, and O. Chiotti, in: *Emerging Issues on Supply Chain Management/ Decentralized Process Management for Inter-Enterprise Collaboration*, edited by B. Sahay (MacMillan India Limited, New Delhi, India, 2004).
11. M.L. Caliusco, M.R. Galli, and O. Chiotti, Ontology and XML-based Specifications for Collaborative B2B Relationships, *CLEI Electronic Journal*, Special Issue of Best Papers presented at JISIC'2003, 7(1), Paper 5 (2004).
12. N. Noy, and D. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford Knowledge Systems Laboratory, Technical Report KSL-01-05, and Stanford Medical Informatics, Technical Report SMI-2001-0880, March 2001.
13. A. Gómez-Pérez, M. Fernández, and A.J. de Vicente, in: *Proceedings of the Workshop on Ontological Engineering/ Towards a Method to Conceptualize Domain Ontologies*, edited by P. van der Vet (Budapest, Hungary, 1996), pp. 41-52.
14. M. Theodorakis, *Contextualization: An Abstraction Mechanism for Information Modeling*, PhD thesis, (Department of Computer Science, University of Crete, Greece, 2001).