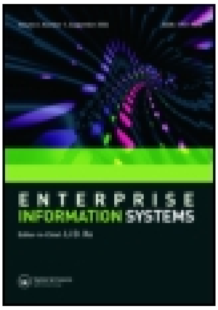

Enterprise Information Systems
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/teis20

An approach to define semantics for BPM systems interoperability

Mariela Rico [a] , María Laura Caliusco [a] , Omar Chiotti [a b] & María Rosa Galli [a b]

[a] CIDISI Research Center, UTN-FRSF, Lavaise 610, S3004EWB, Santa Fe, Argentina

[b] INGAR-UTN-CONICET, Avellaneda 3657, S3002GJC, Santa Fe, Argentina

Version of record first published: 14 Mar 2013.

To cite this article: Mariela Rico , María Laura Caliusco , Omar Chiotti & María Rosa Galli (2013): An approach to define semantics for BPM systems interoperability, Enterprise Information Systems, DOI:10.1080/17517575.2013.767381

To link to this article: http://dx.doi.org/10.1080/17517575.2013.767381

Enterprise Information Systems, 2013
http://dx.doi.org/10.1080/17517575.2013.767381

Taylor & Francis
Taylor & Francis Group

An approach to define semantics for BPM systems interoperability

Mariela Rico[a]*, María Laura Caliusco[a], Omar Chiotti[a,b] and María Rosa Galli[a,b]

[a]*CIDISI Research Center, UTN-FRSF, Lavaise 610, S3004EWB Santa Fe, Argentina;* [b]*INGAR-UTN-CONICET, Avellaneda 3657, S3002GJC Santa Fe, Argentina*

(*Received 24 February 2012; final version received 13 January 2013*)

This article proposes defining semantics for Business Process Management systems interoperability through the ontology of Electronic Business Documents (EBD) used to interchange the information required to perform cross-organizational processes. The semantic model generated allows aligning enterprise's business processes to support cross-organizational processes by matching the business ontology of each business partner with the EBD ontology. The result is a flexible software architecture that allows dynamically defining cross-organizational business processes by reusing the EBD ontology.

For developing the semantic model, a method is presented, which is based on a strategy for discovering entity features whose interpretation depends on the context, and representing them for enriching the ontology. The proposed method complements ontology learning techniques that can not infer semantic features not represented in data sources. In order to improve the representation of these entity features, the method proposes using widely accepted ontologies, for representing time entities and relations, physical quantities, measurement units, official country names, and currencies and funds, among others. When the ontologies reuse is not possible, the method proposes identifying whether that feature is simple or complex, and defines a strategy to be followed. An empirical validation of the approach has been performed through a case study.

Keywords: BPM system; cross-organizational business process; context; semantics; ontology; method

1. Introduction

Business Process Management (BPM) systems of enterprises interact among them to perform cross-organizational business processes (Kalogeras et al. 2006). Interactions are carried out by interchanging messages usually orchestrated by appropriate interaction protocols. Information required to perform cross-organizational processes is interchanged through Electronic Business Documents (EBD) attached to messages (Waldt and Drummond 2006; RosettaNet nd; Rowell and Feblowitz 2002). They are XML-based standard documents where semantics of information included in them can be defined through ontologies (EBD ontologies) (Izza 2009). Assuming a three-layered architecture for ontology-based BPM systems (Jung 2009), business processes of each business partner have to be aligned for achieving semantic interoperability between heterogeneous BPM systems. This requires finding correspondences between the business ontology of each business partner and the EBD ontology. These correspondences are obtained by performing a matching process. This process has to deal with heterogeneities caused by

*Corresponding author. Email: micro@santafe-conicet.gov.ar

both different terminology and knowledge structures (Jung 2011), and different contexts in which entities are considered.

Some entity features affect an entity interpretation depending on the context in which it is considered. Then, matching two ontologies that were defined in different contexts can be difficult if not impossible when these entity features are not represented in them. In order to overcome this drawback, these entity features have to be represented. Representing entity features leads to distinguish between simple entity features, which can be measurable or non-measurable, and complex entity features. When an entity is modelled considering different features, sub-contexts and overlapping contexts can arise (Rico et al. 2007). Ontology design patterns are considered a promising contribution to ontology development (Hammar and Sandkuhl 2010). Specifically, three patterns were proposed to represent features of independent entities (Egaña et al. 2008). Since these patterns were designed to model features in the biological domain, its application in the BPM system interoperability area considered in this article does not result in a valid model.

Ontology development is a time and resources demanding task; consequently, techniques for building an ontology in a semiautomatic fashion from relational schemas (e.g. Kashyap 1999; Rubin et al. 2002; Lehmann and Hitzler 2008; Cerbah 2008) and semistructured ones (e.g. Doan et al. 2000; Delteil et al. 2002; Benz 2007) were proposed. A review of these techniques, called ontology learning, can be found in Gómez-Pérez and Manzano-Macho (2003) and Drumond and Girardi (2008).

This article presents an EBD ontology-based approach to define semantics for BPM systems interoperability. Taking advantage of ontology learning techniques from semistructured data sources, this approach proposes a method based on a strategy for identifying and representing entity features whose interpretation depends on the context in which they are considered. The method facilitates finding correspondences between the business ontology of each business partner and EBD ontologies. The approach, which integrates context notion in a more simple way than that presented in Analyti et al. (2007), considers sub-contexts and overlapping contexts, and proposes a set of design principles in order to overcome drawbacks of ontology design patterns in modelling complex entity features and simple entity features, which can be measurable or non-measurable.

The article is organized as follows: Section 2 discusses related work. Section 3 defines terminology used. Section 4 presents a method for developing a domain ontology (EBD ontology) in a semiautomatic way from semistructured data sources. Section 5 presents an empirical validation of the method through a case study. Finally, Section 6 is devoted to conclusions.

2. Related work

In database and global information system area, information interchange by using a notion of context was addressed. In order to facilitate semantic interoperability between heterogeneous information systems, the use of semantic values as interchange unit was proposed (Sciore et al. 1994). In this approach, source and receiver systems keep locally contextual information for their data, and a context mediator converts data using a shared ontology that specifies terminology mappings between contextual information from different systems. A similar work was proposed for multi-database environments (Goh et al. 1999).

Semantic heterogeneity problem in global information systems was also addressed (Kashyap and Sheth 1998). The approach considers each repository has a domain specific ontology, whose elements are used to construct contextual descriptions that define the meaning of terms used in intentional descriptions. Since contextual descriptions are constructed from different ontologies, terminological relations (e.g. synonyms, hyponyms, and hypernyms) between terms across ontologies are used in order to achieve semantic interoperability. Each information system exports to the global information system a global object corresponding to objects it manages. The context of a query on the global information system is constructed by using contextual descriptions and compared with contexts of exported objects for obtaining an answer.

Previous approaches result in a context model that works well in a domain where it is possible to build a single shared ontology and relate it to multiple contexts. However, arriving at a global shared ontology for semantic interoperability among heterogeneous BPM systems, as that involved in cross-organizational processes, is sometimes impossible. Furthermore, approaches do not consider the effect of sub-contexts and overlapping contexts on entity interpretation.

A generic notion of context could be used as an abstraction mechanism for information modelling (Analyti et al. 2007). It defines a context c as a set of real-world objects, where each object is associated with a set of names and (possibly) with a reference to another context c' that contains further information about the object. Different references of an object (from different contexts) signify different partial representations of the same object. Thus, this notion of context distinguishes between objects and contexts, and supports nesting of contexts and context overlapping. When simple and complex entity features are modelled following this approach, a complex model with excessive number of incorporated contexts is obtained. Drawbacks are twofold: a complex model results in a complex implementation task and a complex information system.

Although different approaches for creating ontology have been proposed (Grüninger and Fox 1995; Uschold and King 1995; Uschold and Grüninger 1996; Kashyap 1999; Maedche and Staab 2001; Noy and McGuinness 2001; Gómez-Pérez et al. 2004; Pinto *et al.* 2004; Sure et al. 2004; Brusa et al. 2008; Suárez-Figueroa and Gómez-Pérez 2009), efficient ontology development continues to be a challenge (Hammar and Sandkuhl 2010). Ontology design patterns that encode best design practices are a promising contribution to this challenge (Gangemi et al. 2007). These patterns were grouped into six families: structural, correspondence, content, reasoning, presentation, and lexical-syntactic patterns (Gangemi and Presutti 2009).

Among several ontology patterns proposed (Blomqvist et al. 2009, 2010; Noy and Rector 2006), there are approaches for modelling descriptive features (Rector 2005), and for representing features and its values, of independent entities (Egaña et al. 2008). These last patterns were called entity-quality, entity-property-quality, and entity-feature-value. Entity-quality and entity-property-quality patterns are intended to model simple, non-measurable entity features, whereas entity-feature-value is intended to model complex entity features. However, simple measurable features are not modelled, and the application of these patterns, designed to model entity features in the biological domain, does not result valid for modelling information interchanged between different contexts required by cross-organizational business processes.

In order to overcome mentioned drawbacks, this article presents an approach to model both complex and simple entity features. Preliminary ideas to represent entity features in a domain ontology were already presented (Rico et al. 2009). In the present work, these ideas were used for building a method to develop models for allowing semantic interoperability between autonomous ontology-based BPM systems. This method allows modelling complex entity features and simple entity features, which can be measurable or non-measurable. Restrictions of existence, cardinality, and closure axioms were introduced in order to guarantee coherence and consistency in modelling complex entity features.

3. Conceptual foundations

Domain is a portion of reality that forms the subject matter of a single field of study, a technology, or a mode of study (Smith et al. 2006).

Context is defined as a set of real-world objects that are of interest (Analyti et al. 2007). Interpretation and representation of an entity and its features depend on the context in which they are considered. For example, an entity is interpreted as 'product' in the context of a packaging industry and as 'package' in the context of a dairy industry.

Semantics for a domain can be represented through a domain ontology. For example, in this article, semantics for an EBD is represented through a domain ontology called EBD ontology.

Ontology elements are:

- **Terms**: words or groups of words representing entities or entity features considered entities in themselves.
- **Properties**: represent entity features that are not considered entities in themselves.
- **Relations**: elements that link together ontology elements.
- **Instances**: refer to individuals. A term and its instances are related by an instance-of relation.
- **Axioms**: represent sentences always true in a domain (Grüber 1993). They are used to represent restrictions about entities (Gómez-Pérez et al. 2004).

4. Method for developing a domain ontology

The objective is to develop a domain ontology (from now on 'ontology') that fulfils use requirements and design criteria, from semistructured data sources (e.g. XML-based EBDs). The proposed method is based on a semiautomatic strategy for discovering and representing entity features, called contextual features, whose interpretation depends on context. The method is composed of five processes (Figure 1): identifying ontology requirements, generating a base ontology, verifying completeness of the ontology, enriching the ontology, and validating the ontology.

4.1. *Process 1: Identifying ontology requirements*

The process objective is to identify requirements the ontology has to satisfy. It is composed of three activities: describe the domain, determine ontology goal and scope, and identify main entities, relations, and features. The output is a specification document containing a domain description, the ontology goal and scope, and a list of entities, their relations, and features.

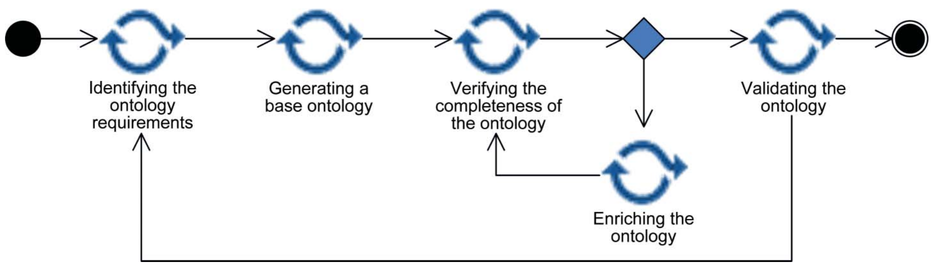

Figure 1. Method processes.

4.1.1. Activity 1.1: Describe the domain

The first task is to define the domain the ontology has to cover (Noy and McGuinness 2001). To this aim, meetings with domain experts allow acquiring the domain knowledge needed to perform activities the method involves. Results will be: description of data sources, scenarios, competency questions the ontology should respond, and involved contexts.

Scenarios can be documented using templates (Brusa et al. 2008). From scenarios, a set of questions that pose demands on the underlying ontology have to be defined. These are 'informal competency questions' that should be formulated following an increasing complexity level (Grüninger and Fox 1995; Uschold and Grüninger 2006).

4.1.2. Activity 1.2: Determine ontology goal and scope

Goal specifies future ontology uses. Particularly, the method proposes to develop an EBD ontology for allowing information interchange between contexts. Users are heterogeneous BPM systems that need the ontology to perform cross-organizational business processes; therefore, from them more detail about goal can be specified.

Scope limits entity representation. To this aim, scenarios and competency questions could be used.

4.1.3. Activity 1.3: Identify main entities, relations, and features

Entities are obtained by exploring outputs of previous activities. Features can be elicited from the knowledge required to understand the meaning of entities in different contexts. In order to distinguish contextual features, the following question should be answered: *Are there any entity features whose meanings could change depending on context?*

4.2. Process 2: Generating a base ontology

A base EBD ontology is generated from the XML-based EBD BPM systems use to interchange information resources. It can be created manually, but the method proposes to use an ontology learning technique in order to decrease the development time. For semistructured data sources such as XML-based EBD, learning techniques that can be used are:

- **Papatheodorou method**, which uses a data mining approach for building taxonomies (Papatheodorou et al. 2002).
- **Volz approach**, which uses a set of rules for capturing semantics (Volz et al. 2005).

The output of Process 2 is a base EBD ontology formally represented in the language the learning technique uses.

4.3. Process 3: Verifying completeness of the ontology

The completeness of the ontology is verified by checking if design principles and use requirements (output of Process 1) are met. The output is a report of how requirements are fulfilled.

If contextual features are not identified by learning techniques, the base EBD ontology can not satisfy use requirements. By analysing informal competency questions, entities, features, and relations needed to answer them can be intuitively identified (Uschold and Grüninger 1996). Then, it has to be identified if they are represented in the ontology; and if such representation is adequate. If this is so, the validation has to be performed (Process 5). Otherwise, the ontology has to be enriched (Process 4). Verification and enrichment processes may require several cycles. Activities this process involves are: identify the representation of entities, and analyze the representation of entities, relations, and features.

4.3.1. Activity 3.1: Identify the representation of entities

Ontology elements have to be analyzed to identify if required entities, relations, and features are represented. Depending on data source model, an entity could be represented by a simple term or a set of ontology elements, and it is also possible that some entities, relations, or features are not represented.

4.3.2. Activity 3.2: Analyze the representation of entities, relations, and features

For identifying representation problems, qualitative approaches can be used (Grüber 1995; Gómez-Pérez 1999; Brewster et al. 2004). Analysis has to be performed following design criteria: clarity, minimal encoding bias, extendibility, coherence, and minimal ontological commitments (Grüber 1995; Sheldon et al. 2003).

Even if the entity representation fulfils the aforementioned criteria, some contextual features could not be represented or be incompletely represented. For improving representation, based on outputs of previous processes, contextual features can be identified. Since not all of them need to be represented, a set of guide questions is proposed (Table 1), whose answers help to identify those to be represented.

As output, a list of entities not represented, and a list of missing contextual features and relations are generated.

4.4. Process 4: Enriching the ontology

Representation of entities is improved according to evaluation results (output of previous process). Process activities are: add missing elements; improve the representation of entities, relations, and features; and designate a bridge term that refers to entity. The output is an enriched ontology.

Table 1. Guide questions.

GQ_1	Are there any unrepresented features of an entity that may be inferred by a person but not by a machine? If the answer is yes, could these entity features be inferred in the wrong way in other contexts different from the considered context? If the answer is yes, these are contextual features that should be represented.
GQ_2	Are the representations and meanings of the contextual features or entities completely represented in the ontology? If the answer is no, they should be represented.
GQ_3	What dimensions are used to specify contextual features? Are they the same regardless of the context in which the features are considered? If the answer is no, are they represented in the ontology? If the answer is no, these dimensions should be represented.

4.4.1. Activity 4.1: Add missing elements

Ontology elements needed to represent missing elements are created following the aforementioned design criteria.

4.4.2. Activity 4.2: Improve the representation of entities, relations, and features

In order to improve the representation of entities, relations, and contextual features, two alternatives are proposed: reusing existing ontologies and applying design principles.

4.4.2.1. Reusing existing ontologies. Integrating ontologies so that they can share and reuse each other's knowledge is beneficial. If a time ontology, for example, has a well-developed time representation, another ontology could use this representation without having to reinvent it (Noy and Hafner 1997). Then, reusing existing and widely accepted ontologies or parts of them should be considered. Examples of such ontologies are:

- TOVE ontologies, which consist of a set of generic core ontologies including: an activity ontology that spans activity, state, time, and causality; a resource ontology; an organization ontology that spans structure, roles, and communication; and a product ontology that includes features, parameters, assemblies, versions, and revisions (Grüninger et al. 2000).
- The OWL-Time ontology for representing most of time entities and relations, i.e., a vocabulary for expressing facts about topological relations among instants, intervals, and events, together with information about durations, dates, and times (Hobbs and Pan 2004).
- The ISO 3166 country codes ontology[1] for representing official country names of ISO 3166.[2]
- A portion of an ontology representing the ISO's currency codes, which are published in the standard ISO 4217:2008, such as the PCS ontology[3] for representing currencies and funds.
- A portion of an ontology representing the international standard ISO 80000, successor of ISO 31, for representing physical quantities and measurement units, such as The United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT).[4]

4.4.2.2. Applying design principles. When it is not possible to reuse an ontology to improve a contextual feature representation, it is necessary to identify if that feature is simple or complex. A simple feature is a quality that does not bear other qualities, and it is associated with a one-dimensional representation (Guizzardi 2005). Thus, in this case, two elements should be added to the ontology:

- A term representing the dimension
- A relation between this term and that representing the simple contextual feature.

Figure 2 shows a simple feature example. Feature country is represented by term CountryDimension, which represents the set of possible values, and relation inCountry between terms CountryDimension and Address.

When a simple feature is measurable its dimension has an associated measurement unit (grams, meters, cm^3, dollars, etc.) that affects its dimension granularity but not its

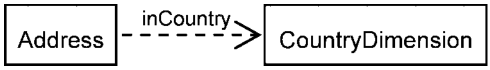

Figure 2. A simple feature example.

structure. In turn, measurement unit is associated with a physical dimension, such as density, length, frequency, mass, time, and speed, among others. Thus, a measurable simple feature should be represented by the following elements:

- A term representing the dimension
- A relation between this term and that representing the simple contextual feature
- A term representing the measurement unit of the dimension
- A relation between this term and that representing the dimension
- A term representing the physical dimension of the measurement unit
- A relation between this term and that representing the measurement unit

Figure 3 shows the representation of feature weight. Term WeightDimension, representing a one-dimensional representation, is related to its measurement unit through relation measuredIn. Term UnitOfMeasure represents the measurement unit of dimension weight, whose possible values are positive real numbers. Term PhysicalDimension represents the physical dimension of the measurement unit, and it is related to term representing the measurement unit through relation hasDimension.

A complex feature is a quality that bears other qualities, and it is associated with a set of integral dimensions that are separable from all other dimensions (Guizzardi 2005). An integral dimension is one in which for assigning a value to an object on one dimension, it is necessary giving it a value on other dimensions. For instance, size can be represented in terms of integral dimensions depth, width, and height. By contrast, weight and depth dimensions are said to be separable. In order to improve the representation of a complex feature, the following elements should be added:

- A term representing the set of integral dimensions
- A relation between this term and that representing the entity
- For each integral dimension, a term representing it and a relation between it and that representing the set of integral dimensions
- For each term representing a measurable, integral dimension, the following elements should be added: a term representing the measurement unit of the integral dimension; a relation between this term and that representing the measurable integral dimension; a term representing the physical dimension of the measurement unit; and a relation between this term and that representing the measurement unit.

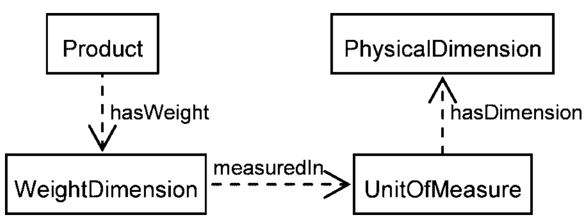

Figure 3. A measurable simple feature example.

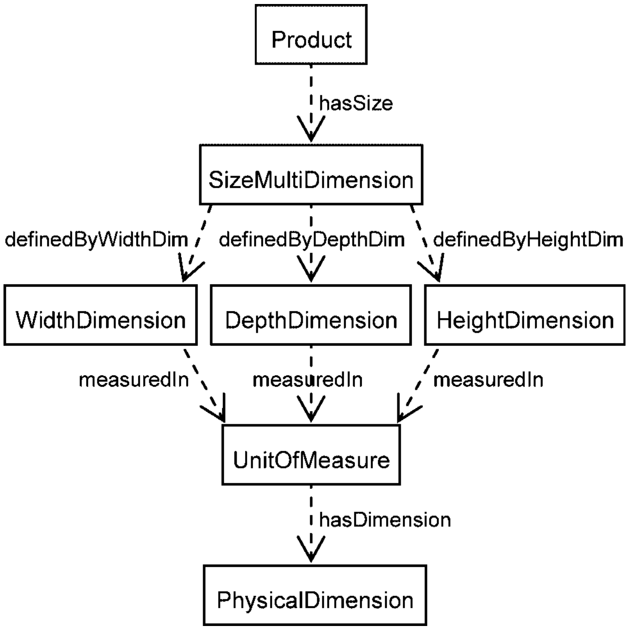

Figure 4. A complex feature example.

Figure 4 shows a complex feature example. Term SizeMultiDimension represents the set of integral dimensions and it is associated with term Product representing entity product through relation hasSize. Terms WidthDimension, DepthDimension, and HeightDimension represent three integral dimensions that compose a set of integral dimensions, and they are associated with term SizeMultiDimension through relations definedByWidthDim, definedByDepthDim, and definedByHeightDim, respectively. In this case, all integral dimensions are measurable and therefore terms representing them are associated with term UnitOfMeasure through relation measuredIn.

Restrictions of existence, cardinality, and closure axioms must be added to ensure that the set of integral dimensions is made by the corresponding integral dimensions. They must be necessary and sufficient conditions. Another restriction to be defined is one requiring that the integral dimensions must be disjoint. For the example in Figure 4, this is:

$$\begin{aligned}\text{SizeMultiDimension} \equiv\ & \exists\ \text{definedByWidthDim.WidthDimension} \sqcap \\ & \forall\ \text{definedByWidthDim.WidthDimension} \sqcap \\ & (\leq 1)\ \text{definedByWidthDim} \sqcap \\ & \exists\ \text{definedByDepthDim.DepthDimension} \sqcap \\ & \forall\ \text{definedByDepthDim.DepthDimension} \sqcap \\ & (\leq 1)\ \text{definedByDepthDim} \sqcap \\ & \exists\ \text{definedByHeightDim.HeightDimension} \sqcap \\ & \forall\ \text{definedByHeightDim.HeightDimension} \sqcap \\ & (\leq 1)\ \text{definedByHeightDim}\end{aligned}$$

$$\begin{aligned}\text{DepthDimension} \sqcap \text{HeightDimension} &\sqsubseteq \bot \\ \text{DepthDimension} \sqcap \text{WidthDimension} &\sqsubseteq \bot \\ \text{WidthDimension} \sqcap \text{HeightDimension} &\sqsubseteq \bot\end{aligned}$$

4.4.3. Activity 4.3: Designate a bridge term that refers to entity

A key aspect to represent is intended uses of an entity in a context, which should be represented by terms, called bridge, that allow linking meanings and representations of an entity in different contexts. Although bridge terms are widely used for crossing ontology domains, in this approach they should also be interpreted as representing contextual features, because intended uses depend on context. Thus, it is necessary to identify if each entity has a term designating its intended uses. If such term is absent, it has to be added. To represent bridge terms, suitable terms of existing vocabularies should be reused. A set of vocabularies for describing common things like people, places, or projects has emerged in the Linked Data community (Heath and Bizer 2011).

Bridge terms should also be related to elements representing entities whose intended uses they represent. An entity could be represented by a single element or a set of elements. In the former case, a relation between that single element and the bridge term should be added. In the latter, the term that best represents an entity should be chosen, and a relation between this term and the bridge term should be added. As that bridge term represents a contextual feature, it should also be related to the term that represents its dimension.

Enriched ontology should be verified again (return to Process 3).

4.5. Process 5: Validating the ontology

Validating means checking if the developed EBD ontology meets uses requirements from the viewpoint of BPM systems involved in the information interchange. Requirement specifications and competency questions can be used as reference frame for validation. Competency questions serve two purposes: for verification, they are used to determine if ontology can answer them; and for validation, it must be checked if competency questions actually posed right questions for ontology purpose (de Almeida Falbo 2004).

Inadequate competency questions must be reformulated, making it necessary to review everything done since requirements identification (Process 1).

If competency questions are formally represented, validation can be performed automatically. A semiautomatic validation can be performed using specific heuristics and human judgments. In order to formalize competency questions, SPARQL (Prud'hommeaux and Seaborne 2008) and OWL-QL (Fikes et al. 2004) could be used. OWL-QL is suitable when queries require carrying out inferences.

5. Validation of the method

5.1. Case study

A case study was used for validation purpose due to it offers the opportunity to study a phenomenon in its own natural setting (Miles and Huberman 1994; Yin 2009).

This case study is based on a collaborative relationship between a packaging industry, called dairy packaging, and a dairy industry, called DairyNY Cooperative. Dairy packaging provides packaging to DairyNY Cooperative. The information each enterprise's BPM system needs to execute the cross-organizational process is interchanged through XML-based EBDs. The objective was to develop a semantic model for allowing semantic interoperability between BPM systems.

5.2. Development of the semantic model

Following the method presented in this article, a domain ontology was developed from each XML-based EDB (data source) used in the cross-organizational process. Main details are described through an example.

5.2.1. Process 1: Identifying ontology requirements

5.2.1.1. Activity 1.1: Describe the domain. Data sources involved were identified. Domain experts collaborated by answering the following questions:

- What domain will the ontology cover?
- What contexts can be identified in the domain?
- What features describe these contexts?

Figure 5 shows an example of information interchanged between BPM systems in order to agree on a replenishment plan. Figure 6 shows EBD syntax and structure, expressed in XML schema (Thompson et al. 2004; Biron and Malhotra 2004).

Data structure and contexts (dairy industry and packaging industry) of BPM systems are different. Then, to achieve semantic interoperability, enterprises have to agree on the meaning of information contained in each XML-based EBD. To this aim, for each EDB an ontology has to be developed.

Based on the information collected, competency questions were derived. For example:

DairyNY Cooperative
110 Main Street - Scotia
New York (12345)
Contact Person: Jane Doe

Replenishment Plan
Nº 122364
Date: 12/08/09

Supplier: Dairy Packaging
Address: 100 Pine Blvd - Norton, OH 44203
Contact Person: John Smith

Horizon: 01/09 – 01/10

PERIOD	PRODUCT				QUANTITY	PRICE
	Id	Trademark	Type	Size		
1/09 – 15/09	20320101	yy	carton	1000	4400	0.40
	20320102			2900	2880	0.41
	20070231		plastic	196	1600	0.45
	20070232			250	1800	0.47
	20070235			1000	6500	0.49
	20320101	zz	carton	1000	2200	0.37
	20320102			2900	8064	0.35
	20070232		plastic	250	1800	0.39
	20070235			1000	6500	0.38
16/09 – 25/09	…	…	…	…	…	…
	…	…	…	…	…	…

Figure 5. Example of the exchanged information to agree on a replenishment plan.

- Given a replenishment plan number, what horizon the replenishment plan considers?
- Who are contact persons?
- What are main features of a product?

```xml
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          targetNamespace="http://www.EBD.org"
          xmlns="http://www.EBD.org"
          elementFormDefault="qualified">
 <xsd:complexType name="Agent">
  <xsd:sequence>
     <xsd:element name="Phone"            type="xsd:string" maxOccurs="unbounded"/>
     <xsd:element name="Email"            type="xsd:string" maxOccurs="unbounded"/>
     <xsd:element name="Fax"              type="xsd:string" maxOccurs="unbounded"/>
     <xsd:element name="Name"             type="xsd:string"/>
     <xsd:element name="MailingAddress"   type="Address"/>
  </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="Address">
  <xsd:sequence>
     <xsd:element name="Country"          type="xsd:string"/>
     <xsd:element name="Zip"              type="xsd:integer"/>
     <xsd:element name="City"             type="xsd:string"/>
     <xsd:element name="Street"           type="xsd:string"/>
     <xsd:element name="State"            type="xsd:string"/>
  </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="Organization">
  <xsd:complexContent>
     <xsd:extension base="Agent">
       <xsd:sequence>
         <xsd:element name="ContactPerson" type="Personnel"/>
       </xsd:sequence>
     </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
 <xsd:complexType name="Personnel">
  <xsd:complexContent>
     <xsd:extension base="Agent">
       <xsd:sequence>
         <xsd:element name="Position"         type="xsd:string"/>
       </xsd:sequence>
     </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
 <xsd:complexType name="Item">
  <xsd:sequence>
      <xsd:element name="PartNumber"       type="xsd:integer"/>
      <xsd:element name="Period"           type="xsd:string"/>
      <xsd:element name="ItemDescription"  type="xsd:string"/>
      <xsd:element name="ItemName"         type="xsd:string"/>
      <xsd:element name="Quantity"         type="xsd:integer"/>
      <xsd:element name="Price"            type="xsd:float"/>
  </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="EBD">
  <xsd:sequence>
     <xsd:element name="Supplier"          type="Organization"/>
     <xsd:element name="Customer"          type="Organization"/>
     <xsd:element name="EBDNumber"         type="xsd:integer"/>
     <xsd:element name="Horizon"           type="xsd:string"/>
     <xsd:element name="EBDDate"           type="xsd:string"/>
     <xsd:element name="EBDName"           type="xsd:string"/>
     <xsd:element name="ItemCollections"   type="Item" maxOccurs="unbounded"/>
  </xsd:sequence>
 </xsd:complexType>
 <xsd:element name="ReplenishmentPlan" type="EBD"/>
</xsd:schema>
```

Figure 6. An XML-based document.

5.2.1.2. Activity 1.2: Determine ontology goal and scope. In this scenario, ontology goal is to describe semantically the information BPM systems interchange to perform the cross-organizational business process whose objective is to agree on a replenishment plan. Initially, ontology scope is limited to answer the following questions:

- What entities compose a replenishment plan?
- What features characterize each replenishment plan entity?
- How are these entities related to each other?

5.2.1.3. Activity 1.3: Identify main entities, relations, and features. Entities involved in a replenishment plan that were identified are: Enterprises, Replenishment Plan, and Items. Item features are: Product, Period, Quantity, and Price. Table 2 shows relations among these entities.

Analyzing replenishment plan information (Figure 5), entity features listed in Table 3 were derived. In order to identify if an entity feature is a contextual feature, the following questions have to be asked: *Does the entity feature interpretation depend on the context in which it is considered? Which are possible interpretations?* Particular questions have arisen. For example:

- Is the beginning and the end of feature Horizon defined by month and year, or by day and month? For example, does the value '01/09' of Horizon represent '1st September' or 'January 2009'?
- What does feature Date mean? Is feature Date defined by day, month, and year or by month, day, and year?
- Does feature Quantity refer to units of products or units of product packages?
- Does feature Price include taxes?

Table 2. Relations among entities.

	Enterprise	Replenishment Plan	Item
Enterprise		isCustomerOf isSupplierOf	supplies requests
Replenishment Plan	isSentBy isReceivedBy		contains
Item	isSoldBy isBoughtBy	isContainedIn	

Table 3. Entity features.

	Contextual features	Non-contextual features
Enterprise	Address	Organization Name Contact Person
Replenishment Plan	Horizon Date Item Description	Number Supplier Customer
Item	Period Product Quantity Price	

5.2.2. Process 2: Generating a base ontology

Applying a learning technique (Volz et al. 2005) to the XML-based EBD (Figure 6), the base EBD ontology shown in Figure 7 was obtained. In this figure, a box represents a term that has a name represented by a label in the top-centre area. Terms have properties represented by rows into the corresponding box. A relation between terms is represented by an arrow and a label representing its name. Arrows with solid lines correspond to hierarchical relations; arrows with dotted lines correspond to particular relations.

5.2.3. Process 3: Verifying completeness of the ontology

In order to check if the base EBD ontology meets uses requirements, it is necessary to identify elements that represent entities, their relations, and features; and then look through this representation to verify the fulfilling of a set of design criteria. This process is performed for each entity identified (Activity 1.3) and described in the following sections.

5.2.3.1. Verifying the representation of entity Enterprise. The supplier and customer roles Enterprises assume are represented by terms Agent and Organization, their properties, and a relation isa between them. A supplier is related to entity Replenishment Plan by means of relation Supplier between terms EBD and Organization, and a customer by means of relation Customer.

According to Table 3, main features of entity Enterprise are: Organization Name, Contact Person, and Address. Organization Name is represented by property Name of term Agent. Contact Person is represented by relation ContactPerson, terms Personnel and Agent, their properties, and a relation isa between them. Address is a contextual feature represented by term Address, its properties, and relation MailingAddress between terms Agent and Address.

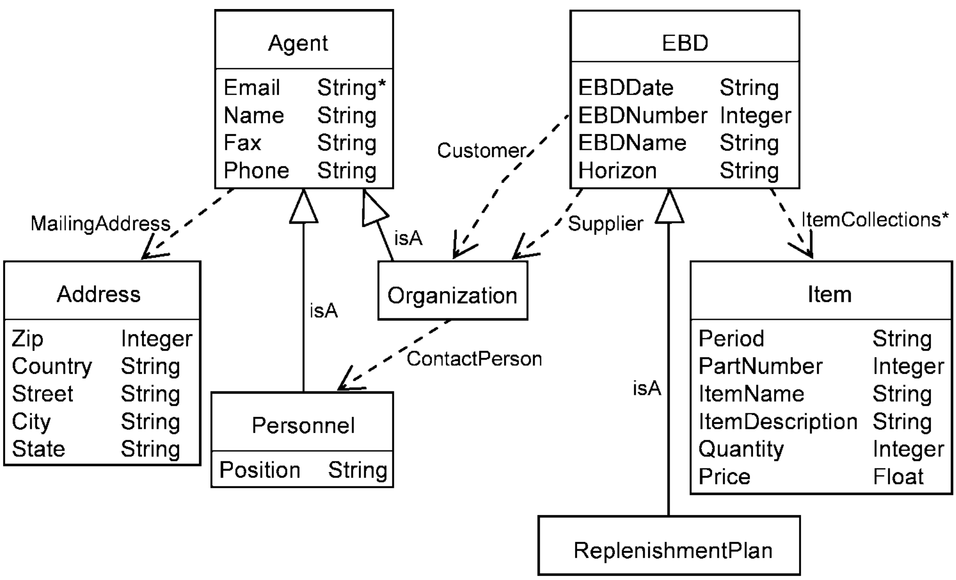

Figure 7. The base EBD ontology.

Term Address represents a mailing address, whose features are: Street, Street Number, Floor, Apartment, City, Post-code, Province or State, and Country. Representation of these features in the base EBD ontology presents some particularities that need to be analysed. Some of them, such as Floor and Apartment are not represented, which is not a problem because the interpretation of the mailing address in different contexts is independent of them. Feature Country is represented by property Country of term Address, whose data type is string. Such representation is inadequate because does not reveal if feature Country refers to either a code or a name. It does neither satisfy the minimal encoding bias criterion nor the minimal ontological commitment criterion. As a result, the representation of feature Country needs to be improved for enabling BPM systems to process it.

5.2.3.2. Verifying the representation of entity Replenishment Plan. Entity Replenishment Plan is represented by terms ReplenishmentPlan, EBD, and Item; their properties; and relations ItemCollections and isa (Figure 7). Terms ReplenishmentPlan and EBD refer to the EBD that contains information about replenishment plans. Term Item represents the document structure.

According to Table 3, the features of entity Replenishment Plan are: Supplier, Customer, Date, Horizon, Number, and Item Description.

Features Supplier and Customer are represented by term Enterprise (Section 5.2.3.1) while Item Description, which describes the features of the Items that are exchanged, is represented by entity Item. Feature Number is represented by property EBDNumber of term EBD, whose data type is integer.

The time period during which the plan is valid is represented by property Horizon of term EBD, whose data type is string. This representation does not reveal if feature Horizon is an amount of time or an interval. Thus, it does neither satisfy the minimal encoding bias criterion nor the minimal ontological commitment criterion. Then, the representation of this feature has to be improved.

Another contextual feature to be improved is Date, which is represented by property EBDDate of term EBD. Since the data type of this property is string, its semantics is not clear.

5.2.3.3. Verifying the representation of entity Item. According to Table 3, features of entity Item are: Period, Product, Quantity, and Price. The period in which items are required is represented by property Period of term Item, whose data type is string. This representation presents the same problems of feature Horizon aforementioned.

Feature Quantity is represented by property Quantity of term Item, whose data type is integer. At this point, some questions arise: what measurement unit is this quantity expressed? will any BPM system make a correct interpretation of it? if there is any possibility of misunderstanding, the measurement unit should be represented.

Analogous to feature Quantity, it can be made the analysis of feature Price that represents the monetary value of each traded product. This feature is represented by property Price of term Item, whose data type is float. In this case, the following question arises: What currency is the price expressed?

Feature Product is represented through properties of term Item: PartNumber, whose data type is integer, and ItemName and ItemDescription, whose data type is string. PartNumber represents Product Id (Figure 5), but ItemName and ItemDescription not represent, at first sight, any of other product features listed in Figure 5: Trademark, Type, and Size. ItemDescription can refer to a product description in natural language. However, property ItemName should be replaced by three terms that represent product features

Trademark, Type, and Size. Then, this product representation does neither satisfy the clarity criterion nor the coherence criterion. A solution is to consider product as an entity and create its own representation.

5.2.4. Process 4: Enriching the ontology

5.2.4.1. Enriching the representation of entity Enterprise. In order to improve the representation of feature Country, it is necessary to distinguish when that feature is an entity in itself and when it is not. For example, feature Country of a mailing address could be considered an entity too, and be represented by means of term Country instead of a property. In this case, a formal relation (Guizzardi 2005) that glues together terms Address and Country is needed.

In order to satisfy the extendibility criterion (Grüber 1995), term Country and its possible values can be imported from an ontology that models the official country names, as given in ISO 3166.[5] Figure 8 shows a portion of the EBD ontology, which contains modifications introduced (Figure 7).

5.2.4.2. Enriching the representation of entity Replenishment Plan. According to Figure 5, feature Horizon should be represented as a calendar interval since it is a time quantity located on time line.

For representing temporal entities, a time ontology can be used (Hobbs and Pan 2004 Ferguson 2003). Basic temporal entities and relations the most of BPM systems require can be provided by the OWL-Time ontology (Hobbs and Pan 2004), which is a simple vocabulary for expressing facts about topological relations among instants, intervals, and events, together with information about durations, dates, and times.

Considering OWL-Time ontology, feature Horizon of entity Replenishment Plan should be represented through a new term derived from term CalendarClockInterval, and linked to term EBD by a formal relation (Figure 9). Feature Date can be improved using a similar strategy.

5.2.4.3. Enriching the representation of entity Item. According to conclusions of Process 3, the representation of features Period, Product, Quantity, and Price of entity Item has to be improved.

The feature Period representation can be improved using a strategy similar to that used for feature Horizon (Figure 9).

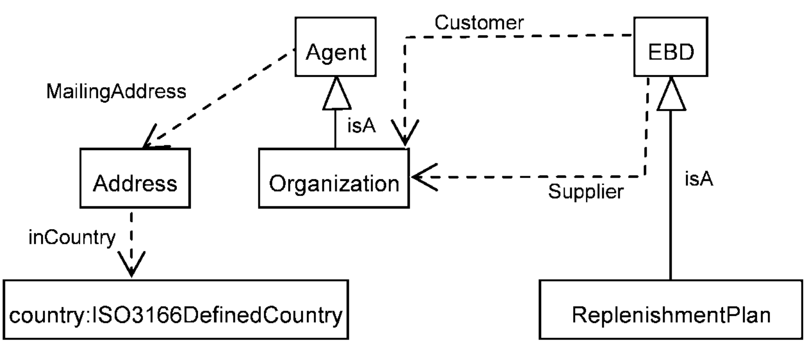

Figure 8. A portion of the enriched EBD ontology representing feature Address.

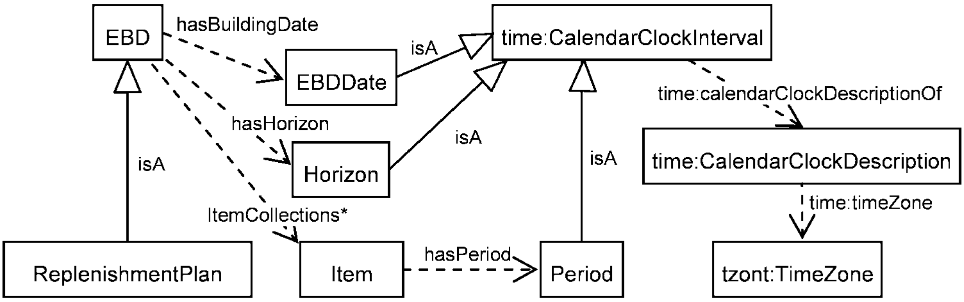

Figure 9. A portion of the enriched EBD ontology representing temporal features.

Feature Product can be represented as an entity so term Product has to be added and related to term Item; and properties PartNumber and ItemDescription have to be properties of term Product. Besides being an entity in itself, Product is a feature of entity Item. As such, it is a complex feature. Product is characterized by the trademark printed in packages, material type with which packages are made, and package capacity. Each of these features has a dimension associated. Term TrademarkDimension represents dimension of feature Trademark. This dimension is not metric, but an enumeration of possible values. In the same way, term TypeDimension represents the type dimension, which is an enumeration of possible values (such as ‘carton’ and ‘plastic’). Finally, term SizeDimension represents dimension of feature Size, which is metric. Since package capacity is an amount associated with a measurement unit, term SizeDimension has a relation with term UnitOfMeasure, which in turn is related to term PhysicalDimension. Three dimensions represented by terms TrademarkDimension, TypeDimension, and SizeDimension are called integral dimensions according to Guizzardi (2005), and compose a multi-dimension represented by term ProductMultiDimension, related to term Product.

In order to ensure that an instance of term ProductMultiDimension is composed of an instance of each of terms TrademarkDimension, TypeDimension, and SizeDimension, the following restrictions must be added:

$$
\begin{aligned}
\text{ProductMultiDimension} \equiv\; & \exists\ \text{definedBySizeDim.SizeDimension} \sqcap \\
& \forall\ \text{definedBySizeDim.SizeDimension} \sqcap \\
& (\leq 1)\ \text{definedBySizeDim} \sqcap \\
& \exists\ \text{definedByTrademarkDim.TrademarkDimension} \sqcap \\
& \forall\ \text{definedByTrademarkDim.TrademarkDimension} \sqcap \\
& (\leq 1)\ \text{definedByTrademarkDim} \sqcap \\
& \exists\ \text{definedByTypeDim.TypeDimension} \sqcap \\
& \forall\ \text{definedByTypeDim.TypeDimension} \sqcap \\
& (\leq 1)\ \text{definedByTypeDim}
\end{aligned}
$$

$$
\begin{aligned}
\text{TrademarkDimension} \sqcap \text{TypeDimension} &\sqsubseteq \bot \\
\text{TrademarkDimension} \sqcap \text{SizeDimension} &\sqsubseteq \bot \\
\text{SizeDimension} \sqcap \text{TypeDimension} &\sqsubseteq \bot
\end{aligned}
$$

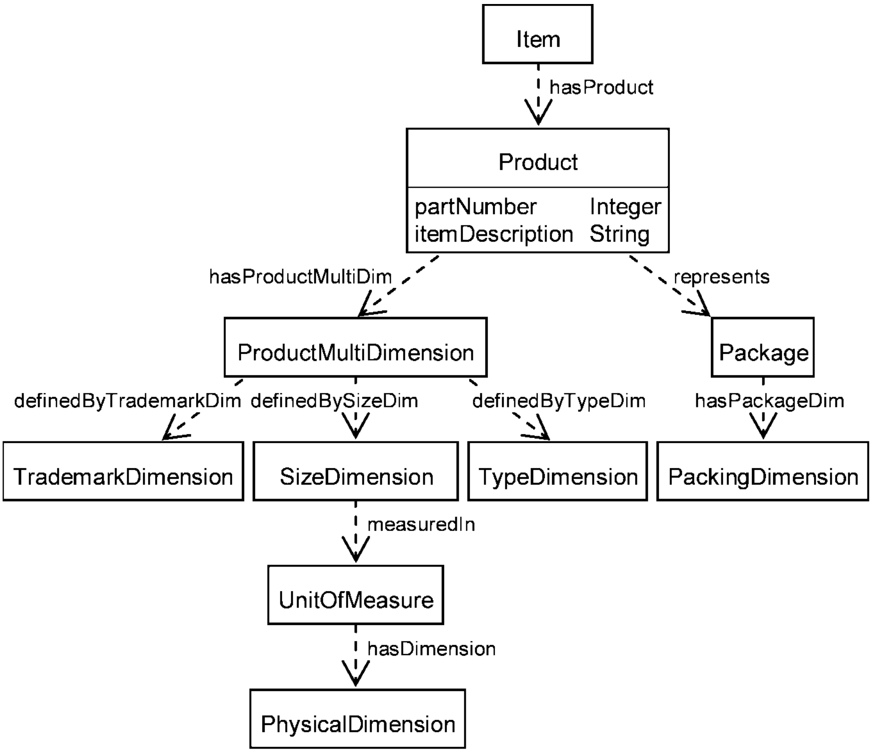

Figure 10. A portion of the enriched EBD ontology representing entity Product.

Figure 10 shows a portion of the EBD ontology with changes made to represent feature Product. The bridge terms Product and Package should be added to represent intended uses of entity Product in both packaging industry and dairy industry contexts. Since Product is already in the ontology, only Package has to be added and related to term Product. As bridge terms represent contextual features, they have to be also related to terms that represent their dimensions. In dairy industry context, packages are associated with a dimension that is an enumeration of possible values.

Feature Quantity is a simple feature associated with a one-dimensional representation, which is isomorphic to the half-line of positive numbers. To represent this feature, three elements should be added to the EBD ontology: a term QuantityDimension; a relation between this term and term UnitOfMeasure; and a relation between terms QuantityDimension and that representing entity Item.

Feature Price is also a simple feature, which should be represented by a formal relation between terms Item and CurrencyDimension. The latter represents the price dimension, whose value is a real number. Monetary unit associated with feature Price is represented through a relation between terms CurrencyDimension and UnitOfMeasure. Figure 11 shows changes made to represent features Quantity and Price.

5.2.5. *Verifying completeness of the enriched ontology*

The enriched EBD ontology has proved verifying requirement specifications and competency questions listed in the ontology requirement document. Conclusions were:

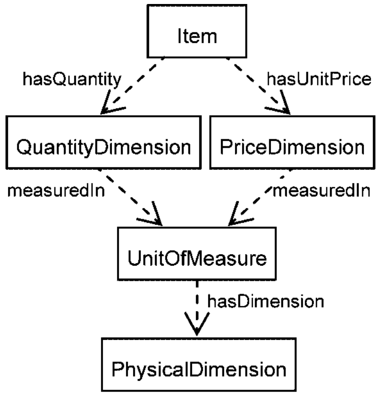

Figure 11. A portion of the enriched EBD ontology representing Quantity and Currency features.

- The EBD ontology is complete for the purpose it was developed.
- The EBD ontology is concise. Property itemName was the only element detected as unnecessary and was removed from the ontology.
- The EBD ontology is clearer.
- The EBD ontology can be extended to model other EBDs.
- Effects of encoding bias and ontological commitments were reduced.

5.2.6. Validating the ontology

In order to validate the enriched EDB ontology, competency questions were formalized using SPARQL query language (Prud'hommeaux and Seaborne 2008). Answers given by the EDB ontology to competency questions were considered acceptable by domain experts.

5.3. Results analysis

The information BPM systems of enterprises (dairy industry and packaging industry) need for executing the cross-organizational process is interchanged through an XML-based EBD. Particularly, the objective of the cross-organizational process addressed in the case study is to agree on a replenishment plan.

Since data structure and contexts of BPM systems are different, enterprises had to agree on the meaning of information contained in the XML-based EBD. To this aim, following the method presented in this article, a base EBD ontology was developed from the XML-based EDB (data source) used in the cross-organizational process by applying a learning technique.

When verified for completeness, the base EBD ontology presented some drawbacks: representations of features Country, Horizon, Date, Quantity, Price, and Product needed to be improved to enable BPM systems to process them. That is, the learning technique could not solve the semantic representation problem the information resource had.

These problems were solved by enriching the ontology through a manual process, following the proposed approach. This allowed: identifying contextual features, relations, and entities not represented in the base EBD ontology; enriching it by adding these missing elements; generating a representation structure to fit the ontology scope; and representing through bridge terms the intended uses of an entity in a context. The latter was performed by linking different meanings and representations of the entity in different contexts.

Even though the learning technique could not solve semantic representation problems the information resource had, it allowed reducing development time because the XML-based EBD was complex to be manually analyzed.

The obtained EBD ontology is simpler than one generated using ontology patterns or the notion of context as an abstraction mechanism (Analyti et al. 2007), which were proposed to develop an ontology for a more general scope. Since the approach proposes a controlled growth of the ontology by including only features, the matching process needs to find correspondences between the business ontology of each business partner and the EBD ontology.

The semantic model developed for BPM systems interoperability allowed aligning business processes to support the collaboration by matching the business ontology of each business partner with the EBD ontology.

The use of an EBD ontology to define semantics for BPM systems interoperability allowed defining a more flexible software architecture than using a global information system with a single shared ontology (Kashyap and Sheth 1998; Goh et al. 1999). Cross-organizational business processes could be defined in a dynamic way by reusing the EBD ontology, extending it to model other EDBs if necessary.

The reuse of existing ontologies proposed by the approach allowed reducing the time for enriching the ontology and obtaining an EBD ontology that can be easily extended.

Eighteen groups of people who had no experience in ontology developments applied the method in a case study based on the definition of XML documents needed to perform a business process to agree on a replenishment plan between manufacturing enterprises. From a modelling point of view, groups had no difficulties in applying the method and results were similar to that previously described for the case study.

6. Conclusions

A contribution of this article is an approach that proposes an EBD ontology-based model to define semantics for BPM systems interoperability. The semantic model generated allows aligning enterprise's business processes to support the collaboration by matching the business ontology of each business partner with the EBD ontology. This results in a software architecture more flexible than a global information system with a single shared ontology, which allows dynamically defining cross-organizational business processes by reusing the EBD ontology, extending it to model other EDBs if necessary.

Other contribution is a method based on a strategy that allows discovering contextual features and representing them for enriching an EBD ontology, complementing ontology learning techniques.

In order to improve the representation of contextual features, the method proposes using widely accepted ontologies, for representing time entities and relations, physical quantities, measurement units, official country names, and currencies and funds, among others. When ontologies reuse is not applicable, the method proposes identifying whether that feature is simple or complex, and proposes a strategy to be followed. By using bridge terms, different meanings and representations of an entity in different contexts are linked. This allows representing intended uses of entities that depend on the context. The EBD ontology obtained is simpler than one generated using ontology patterns or the notion of context as an abstraction mechanism.

The proposed method is useful to generate ontologies from semistructured data sources used for allowing interoperability between applications, such as the Data Package conceived in the DIMP approach (Wang and Xu 2012).

Finally, this work contributes to enterprises' seamless exchange of digital information, which has been stated as an unsatisfied requirement in Jardim-Goncalves et al. (2013).

Acknowledgements

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.

Notes

1. http://www.daml.org/2001/09/countries/iso-3166-ont
2. http://www.iso.org/iso/english_country_names_and_code_elements
3. http://www.ifpi.com/pcs/
4. http://www.unece.org/cefact/codesfortrade/codes_index.htm
5. http://www.iso.org/iso/english_country_names_and_code_elements

References

Analyti, A., M. Theodorakis, N. Spyratos, and P. Constantopoulos. 2007. "Contextualization as an Independent Abstraction Mechanism for Conceptual Modeling." *Information Systems* 32 (1): 24–60.

Benz, D. 2007. "Collaborative Ontology Learning". Master's thesis, Albert-Ludwigs-Universität Freiburg, Department of Computer Science, Germany.

Biron, P. V., and A. Malhotra. 2004. *XML Schema Part 2: Datatypes*. 2nd ed. W3C Recommendation October 28, The World Wide Web Consortium. Available from: http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/datatypes.html

Blomqvist, E., V. K. Chaudhri, O. Corcho, V. Presutti, and K. Sandkuhl. eds. 2010. *Proceedings of the 2nd International Workshop on Ontology Patterns (WOP2010), Collocated with the 9th International Semantic Web Conference (ISWC-2010)*. Shanghai, China, November 8, Vol. 671 of CEUR Workshop Proceedings. CEUR-WS.org

Blomqvist, E., K. Sandkuhl, F. Scharffe, and V. Svatek. eds. 2009. *Proceedings of the Workshop on Ontology Patterns (WOP 2009)*. Washington DC, USA, October 25, Vol. 516 of CEUR Workshop Proceedings. CEUR-WS.org

Brewster, C., H. Alani, S. Dasmahapatra, and Y. Wilks. 2004. "Data Driven Ontology Evaluation." In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004*, Lisbon, Portugal, May 26–28, 164–168. Paris: European Language Resources Association.

Brusa, G., M. L. Caliusco, and O. Chiotti. 2008. "Towards Ontological Engineering: A Process for Building a Domain Ontology from Scratch in Public Administration." *Expert Systems* 25 (5): 484–503.

Cerbah, F. 2008. "Learning Highly Structured Semantic Repositories from Relational Databases: The RDBToOnto Tool." In *The Semantic Web: Research and Applications*, edited by S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis. Vol. 5021 of LNCS, 777–781. Berlin/Heidelberg: Springer.

de Almeida Falbo, R. 2004. "Experiences in Using a Method for Building Domain Ontologies." In *Proceedings of the 16th International Conference on Software Engineering and Knowledge Engineering, SEKE'2004. International Workshop on Ontology in Action, OIA'2004*, 474–477. Banff, Alberta, Canada, June 21. Skokie, IL: Knowledge Systems Institute Graduate School.

Delteil, A., C. Faron, and R. Dieng. 2002. "Building Concept Lattices by Learning Concepts from RDF Graphs Annotating Web Documents." In *Conceptual Structures: Integration and Interfaces. 10th International Conference on Conceptual Structures, ICCS 2002.* Borovets, Bulgaria, July 15–19. *Proceedings*, edited by U. Priss, D. Corbett, and G. Angelova. Vol. 2393 of LNCS, 191–204. Berlin/Heidelberg: Springer.

Doan, A., P. Domingos, and A. Y. Levy. 2000. "Learning Source Descriptions for Data Integration." In *Proceedings of the Third International Workshop on the Web and Databases, WebDB 2000, in Conjunction with ACM PODS/SIGMOD 2000. Informal Proceedings*, 81–86. Dallas, Texas, May 18–19. Florham Park, NJ: AT&T Labs Research.

Drumond, L., and R. Girardi. 2008. "A Survey of Ontology Learning Procedures." In *Proceedings of the 3rd Workshop on Ontologies and Their Applications,* Vol. 427 of CEUR Workshop Proceedings, Salvador, Bahia, Brazil, October 26. CEUR-WS.org.

Egãna, M., A. Rector, R. Stevens, and E. Antezana. 2008. "Applying Ontology Design Patterns in Bio-ontologies." In *Knowledge Engineering: Practice and Patterns. 16th International Conference, EKAW 2008.* Acitrezza, Italy, September 29–October 2. *Proceedings*, edited by A. Gangemi and J. Euzenat. Vol. 5268 of LNCS, 7–16. Berlin/Heidelberg: Springer.

Ferguson, G. 2003. DAML-time homepage [online]. Accessed February 13, 2012. Available from: http://www.cs.rochester.edu/_ferguson/daml/

Fikes, R., P. Hayes, and I. Horrocks. 2004. "OWL-QL. A Language for Deductive Query Answering on the Semantic Web." *Web Semantics: Science, Services and Agents on the World Wide Web* 2 (1): 19–29.

Gangemi, A., A. Gómez-Pérez, V. Presutti, and M. C. Suarez-Figueroa. 2007. "Towards a Catalog of OWL-based Ontology Design Patterns." In *Actas de la XII conferencia de la Asociación Espãnola para la Inteligencia Artificial (CAEPIA – TTIA 2007)*, Salamanca, Espãna, November 12–16.

Gangemi, A., and V. Presutti. 2009. "Ontology Design Patterns: International Handbooks on Information Systems." In *Handbook on Ontologies*, edited by S. Staab and R. Studer. 2nd ed. 221–243. Berlin/Heidelberg: Springer.

Goh, C. H., S. Bressan, S. E. Madnick, and M. Siegel. 1999. "Context Interchange: New Features and Formalisms for the Intelligent Integration of Information." *ACM Transactions on Information Systems* 17 (3): 270–293.

Gómez-Pérez, A. 1999. "Evaluation of Taxonomic Knowledge in Ontologies and Knowledge Bases." In *Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management, KAW'99*, edited by B. R. Gaines, R. Kremer, and M. A. Musen. Vol. 2, 6.8.1–6.8.18. Banff, Alberta, Canada, October 16–21. SRDG, University of Calgary.

Gómez-Pérez, A., M. Fernández-López, and O. Corcho. 2004. *Ontological Engineering*. 2nd ed. *Advanced Information and Knowledge Processing*. New York: Springer London.

Gómez-Pérez, A., and D. Manzano-Macho. 2003. "A Survey of Ontology Learning Methods and Techniques." Deliverable 1.5, OntoWeb Consortium Project number IST-2000-29243. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.3714&rep=rep1&type=pdf

Grüber, T. R. 1993. "A Translation Approach to Portable Ontology Specification." *Knowledge Acquisition* 5 (2): 199–220.

Grüber, T. R. 1995. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing." *International Journal of Human-Computer Studies* 43 (5–6): 907–928.

Grüninger, M., K. Atefi, and M. S. Fox. 2000. "Ontologies to Support Process Integration in Enterprise Engineering." *Computational and Mathematical Organization Theory* 6 (4): 381–394.

Grüninger, M., and M. S. Fox. 1995. "Methodology for the Design and Evaluation of Ontologies." In *Workshop on Basic Ontological Issues in Knowledge Sharing at the Fourteenth International Joint Conference on Artificial Intelligence*, *IJCAI-95*, edited by D. Skuce, 6.1–6.10. Montréal, Québec, Canada, August 20–25.

Guizzardi, G. 2005. "Ontological Foundations for Structural Conceptual Models." PhD diss. University of Twente, Enschede, The Netherlands Telematica Instituut Fundamental Research Series. Vol. 015. Telematica Instituut.

Hammar, K., and K. Sandkuhl. 2010. "The State of Ontology Pattern Research: A Systematic Review of ISWC, ESWC and ASWC 2005–2009." In *Proceedings of the 2nd International Workshop on Ontology Patterns (WOP2010), Collocated with the 9th International Semantic Web Conference (ISWC-2010)*, edited by E. Blomqvist, V. K. Chaudhri, O. Corcho, V. Presutti, and K. Sandkuhl. Vol. 671 of CEUR Workshop Proceedings, 5–17. Shanghai, China, November 8. CEUR-WS.org.

Heath, T., and C. Bizer. 2011. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web: Theory and Technology Vol. 1. San Rafael, CA: Morgan & Claypool.

Hobbs, J. R., and F. Pan. 2004. "An Ontology of Time for the Semantic Web." *ACM Transactions on Asian Language Information Processing (TALIP)* 3 (1): 66–85.
Izza, S. 2009. "Integration of Industrial Information Systems: From Syntactic to Semantic Integration Approaches." *Enterprise Information Systems* 3 (1): 1–57.
Jardim-Goncalves, R., A. Grilo, C. Agostinho, F. Lampathaki, and Y. Charalabidis. 2013. "Systematisation of Interoperability Body of Knowledge: The Foundation for Enterprise Interoperability as a Science." *Enterprise Information Systems* 7 (1): 7–32.
Jung, J. J. 2009. "Semantic Business Process Integration Based on Ontology Alignment." *Expert Systems with Applications* 36 (8): 11013–11020.
Jung, J. J. 2011. "Service Chain-based Business Alliance Formation in Service-oriented Architecture." *Expert Systems with Applications* 38 (3): 2206–2211.
Kalogeras, A. P., J. V. Gialelis, C. E. Alexakos, M. J. Georgoudakis, and S. A. Koubias. 2006. "Vertical Integration of Enterprise Industrial Systems Utilizing Web Services." *IEEE Trans. Industrial Informatics* 2 (2): 120–128.
Kashyap, V. 1999. "Design and Creation of Ontologies for Environmental Information Retrieval." In *Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management, KAW'99*, edited by B. R. Gaines, R. Kremer, and M. A. Musen, 5.4.1–5.4.18. Banff, Alberta, Canada, October 16–21. SRDG, University of Calgary.
Kashyap, V., and A. Sheth. 1998. "Semantic Heterogeneity in Global Information Systems: The Role of Metadata, Context and Ontologies." In *Cooperative Information Systems: Trends and Directions*, edited by M. P. Papazoglou and G. Schlageter, 139–178. San Diego, CA: Academic Press.
Lehmann, J., and P. Hitzler. 2008. "A Refinement Operator Based Learning Algorithm for the ALC Description Logic." In *Inductive Logic Programming*, edited by H. Blockeel, J. Ramon, J. Shavlik, and P. Tadepalli, 147–160. Vol. 4894 of LNCS, Corvallis, OR. Berlin/Heidelberg: Springer.
Maedche, A., and S. Staab. 2001. "Ontology Learning for the Semantic Web." *IEEE Intelligent Systems* 16 (2): 72–79.
Miles, M. B., and A. M. Huberman. 1994. *Qualitative Data Analysis: An Expanded Sourcebook*. 2nd ed. Thousand Oaks, CA: Sage.
Noy, N. F., and C. D. Hafner. 1997. "The State of the Art in Ontology Design: A Survey and Comparative Review." *AI Magazine* 18 (3): 53–74.
Noy, N. F., and D. L. McGuinness. 2001. "Ontology Development 101: A Guide to Creating Your First Ontology." Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory. Available from: http://ksl.stanford.edu/people/dlm/papers/ontologytutorial-noy-mcguinness.pdf
Noy, N., and A. Rector. 2006. "Defining N-ary Relations on the Semantic Web." W3C Working Group Note April 12, The World Wide Web Consortium. Available from: http://www.w3.org/TR/swbp-n-aryRelations/
Papatheodorou, C., A. Vassiliou, and B. Simon. 2002. "Discovery of Ontologies for Learning Resources Using Word-based Clustering. In *Proceedings of World Conference on Educational Multimedia*, *Hypermedia and Telecommunications*, *ED-MEDIA 2002*, edited by P. Barker and S. Rebelsky, 1523–1528. Denver, CO, June 24–29. Jacksonville, FL: AACE.
Pinto, H. S., S. Staab, and C. Tempich. 2004. "DILIGENT: Towards a Fine-grained Methodology for Distributed, Loosely-controlled and Evolving Engineering of Ontologies." In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI)*, edited by R. López de Mántaras and L. Saitta, 393–397. Amsterdam: IOS Press.
Prud'hommeaux, E., and A. Seaborne. 2008. "SPARQL Query Language for RDF." W3C Recommendation January 15, The World Wide Web Consortium. Available from: http://www.w3.org/TR/rdf-sparql-query/
Rector, A. 2005. "Representing specified values in OWL: 'value partitions' and 'value sets'". W3C Working Group Note May 17, The World Wide Web Consortium. Available from: http://www.w3.org/TR/swbp-specified-values/
Rico, M., M. L. Caliusco, O. Chiotti, and M. R. Galli. 2007. "Considering Subcontexts and Multiple Contexts for Defining Semantic Relations between Complex Domains." In *XXXIII conferencia latinoamericana de informática*, *CLEI 2007*, San José, Costa Rica, October 9–12.
Rico, M., M. L. Caliusco, M. L. Taverna, O. Chiotti, and M. R. Galli. 2009. "A Method for Making Contextual Features Explicit in Domain Ontologies." In *XXXV latin-american conference on informatics*, *CLEI 2009*, Pelotas, RS, Brazil, September 22–25.

RosettaNet, nd. RosettaNet standards [online]. Accessed February 13, 2012. Available from: http://www.rosettanet.org/Standards/RosettaNetStandards/tabid/473/Default.aspx

Rowell, M., and M. Feblowitz. 2002. "OAGIS 8.0 Design Document Version 1.1." Document 20021024-1, Open Applications Group, Inc. Available from: http://www.oagi.org/oagi/downloads/designdocument/OAGIS 8.0 Design Document 011.pdf

Rubin, D. L., M. Hewett, D. E. Oliver, T. E., Klein, and R. B. Altman. 2002. "Automatic Data Acquisition into Ontologies from Pharmacogenetics Relational Data Sources Using Declarative Object Definitions and XML." In *Proceedings of the Pacific Symposium on Biocomputing*, edited by R. Altman, A. Dunker, L. Hunter, K. Lauderdale, and T. Klein, 88–99. Stanford, CA: Stanford University.

Sciore, E., M. Siegel, and A. Rosenthal. 1994. "Using Semantic Values to Facilitate Interoperability among Heterogeneous Information Systems." *ACM Transactions on Database Systems* 19 (2): 254–290.

Sheldon, F. T., M. T. Elmore, and T. E. Potok. 2003. "An Ontology-based Software Agent System Case Study." In *Proceedings of the International Conference on Information Technology: Computers and Communications, ITCC '03*, 500–506. Washington, DC: IEEE Computer Society.

Smith, B., W. Kusnierczyk, D. Schober, and W. Ceusters. 2006. "Towards a Reference Terminology for Ontology Research and Development in the Biomedical Domain." In *Proceedings of the Second International Workshop on Formal Biomedical Knowledge Representation: "Biomedical Ontology in Action" (KR-MED 2006), Collocated with the 4th International Conference on Formal Ontology in Information Systems (FOIS-2006)*, edited by O. Bodenreider, 57–65. Vol. 222 of CEUR Workshop Proceedings, Baltimore, Maryland, November 8. CEUR-WS.org.

Suárez-Figueroa, M. C., and A. Gómez-Pérez. 2009. "NeOn Methodology for Building Ontology Networks: A Scenario-based Methodology." In *Proceedings of the International Conference on Software, Services and Semantic Technologies (S3T 2009)*, edited by D. Dicheva, R. Nikolov, and E. Stefanova, 160–167. Sofia: Sofia University

Sure, Y., S. Staab, and R. Studer. 2004. "On-to-knowledge Methodology. International Handbooks on Information Systems." In *Handbook on Ontologies*, edited by S. Staab and R. Studer, Chap. 6, 117–132. Berlin/Heidelberg: Springer.

Thompson, H. S., D. Beech, M. Maloney, and N. Mendelsohn. 2004. *XML Schema Part 1: Structures*. 2nd ed. W3C Recommendation October 28, The World Wide Web Consortium. Available from: http://www.w3.org/TR/xmlschema-1/

Uschold, M., and M. Grüninger. 1996. "Ontologies: Principles, Methods and Applications." *Knowledge Engineering Review* 11 (2): 93–155.

Uschold, M., and M. King 1995. "Towards a Methodology for Building Ontologies." In *Workshop on Basic Ontological Issues in Knowledge Sharing at the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI-95*, edited by D. Skuce. Montréal, Québec, Canada, August 20–25. Edinburgh: Artificial Intelligence Applications, Institute University of Edinburgh. Also available as AIAI-TR-183 from AIAI, The University of Edinburgh.

Volz, R., D. Oberle, S. Staab, and R. Studer. 2005. "OntoLiFT Prototype." Deliverable 11, WonderWeb: Ontology Infrastructure for the Semantic Web IST Project 2001-33052. Available from: http://wonderweb.semanticweb.org/deliverables/documents/D11.pdf

Waldt, D., and R. Drummond. 2006. "EBXML. The Global Standard for Electronic Business. Presentation, Organization for the Advancement of Structured Information Standards." Available from: http://www.xml.org//sites/www.xml.org/files/waldt ebxml global standard ebusiness.pdf

Wang, X. V., and X. W. Xu. 2012. "DIMP: An Interoperable Solution for Software Integration and Product Data Exchange." *Enterprise Information Systems* 6 (3): 291–314.

Yin, R. K. 2009. *Case Study Research: Design and Methods*. 4th ed. Applied Social Research Methods. Vol. 5. Thousand Oaks, CA: Sage.