



Local ontologies for semantic interoperability in supply chain networks

Milan Zdravković, Miroslav Trajanović, Hervé Panetto

► To cite this version:

Milan Zdravković, Miroslav Trajanović, Hervé Panetto. Local ontologies for semantic interoperability in supply chain networks. 13th International Conference on Enterprise Information Systems, ICEIS 2011, Jun 2011, Beijing, China. pp.22-31, <10.5220/0003416500220031>. <hal-00583863>

HAL Id: hal-00583863

<https://hal.science/hal-00583863v1>

Submitted on 30 Apr 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

LOCAL ONTOLOGIES FOR SEMANTIC INTEROPERABILITY IN SUPPLY CHAIN NETWORKS

Milan Zdravković, Miroslav Trajanović

*Laboratory for Intelligent Production Systems, University of Niš, Faculty of Mechanical Engineering, Aleksandra
Medvedeva 14, Niš, Serbia
milan.zdravkovic@masfak.ni.ac.rs, traja@masfak.ni.ac.rs*

Hervé Panetto

*Research Centre for Automatic Control (CRAN – UMR 7039), Nancy-Université, CNRS, Nancy, France
Herve.Panetto@cran.uhp-nancy.fr*

Keywords: Ontology, Enterprise Interoperability, Supply Chain Management, SCOR.

Abstract: Most of the issues of current supply chain management practices are related to the challenges of interoperability of relevant enterprise information systems (EIS). In this paper, we present the ontological framework for semantic interoperability of EISs in supply chain networks, based on Supply Chain Operations Reference (SCOR) model, its semantic enrichment and mappings with relevant enterprise conceptualizations. In order to introduce the realities of the enterprises into this framework, namely their models, we define and implement the approach to generation of local ontologies, based on the databases of their EISs. Also, we discuss on the translation between semantic and SQL queries, a process in which implicit semantics of the EIS's databases and explicit semantics of the local ontologies become inter-related.

1 INTRODUCTION

Despite the advances in the relevant research areas and rapidly growing demand for flexible, customized production, manufacturing supply chains are still primarily focused on a cost reduction as a key aspect of collaboration. The fact that supplier relationship management contributes largely to the overall costs of the supply chains' final products has great impact to their configuration-related decisions. For example, manufacturers tend to reduce the number of suppliers. Moreover, relationships are dyadic – rarely expanded to include vendors' vendors and customers' customers. Also, high level of integration is required in order to reduce costs – manufacturers tend to view their suppliers as extensions of themselves. Traditional approach to supply chains' configuration may have negative impact to their performance. First, high-speed, low-cost supply chains are often unable to respond efficiently to unexpected structural changes in (customized) demand or supply. Second, high level of integration reduces flexibility of small and

medium enterprises, main constituents of the lower levels of supply chains. Third, investments in technical framework for enterprise integration, which could maximize the efficiency and productivity, cannot be returned in a short term. Furthermore, starting collaboration in such traditional settings is reactive and not proactive decision. Namely, relationship establishment or development is motivated by the internal, rather than external factors: complexity and volume of supply relationships, potential for cost reduction (Lamber et al., 2006), high frequency of transactions between parties (Jespersen and Larse, 2006), degree of asset specificity (Williamson, 1985), etc.

In a response to the issues of static and integrated architecture of the supply chain, a notion of virtual enterprise has been introduced and widely discussed in academic community. Virtual enterprise is a temporary network of independent enterprises, who join together quickly to exploit fast-changing opportunities and then dissolve (Browne and Zhang, 1999). It is characterized by a short-living appearance of a supply chain, capable to produce

low volume of high variety of products, by drawing from the loosely-coupled, heterogeneous environment of available competences, capabilities and resources, sometimes referred to as Virtual Breeding Environment (Sánchez et al., 2005). Paradigms of virtual enterprises and their breeding environment are based on the capability of an enterprise to configure or reconfigure quickly, according to the circumstances of the market, often not known in advance, or even in the moment of configuration. Hence, efficiency and effectiveness of this joint endeavour depends on the interoperability of enterprises, rather than their integration. The main prerequisite for achievement of interoperability of the loosely coupled systems is to maximize the amount of semantics which can be utilized and make it increasingly explicit (Obrst, 2003), and consequently, to make the systems semantically interoperable.

In this paper, we discuss on the notion of semantic interoperability in supply chain networks, namely the overall architecture of the enterprise information systems environment and corresponding ontological framework. One of the greatest challenges in building this framework is related to the implicitness of semantics of the enterprises' realities. In our approach to face this challenge, we assume that: 1) these realities are represented by the corresponding enterprise information systems (EIS), and 2) enterprise message models (crucial for flexible economic integration) are based on EISs' data models, represented implicitly in their databases. The proposed approach aims at making this representation - explicit.

Our approach to semantic interoperability in supply networks assumes fragmentation of the problem into three inter-related areas: a) formal model of supply chain, b) enterprise semantics (body of knowledge), and c) local semantics.

Formal model of supply chain (Zdravković et al., 2010) builds upon a widely adopted supply chain process reference model – SCOR (Stewart, 1997). It is represented on two layers of abstraction. First layer models implicit semantics of SCOR elements and stores actual knowledge on supply chain operations. Second layer represents SCOR's semantic enrichment - it identifies common enterprise notions, maps those to SCOR entities and classifies them into more general inter-related concepts. Both layers are then represented by OWL models – SCOR-KOS and SCOR-FULL. This approach is shortly described in section 2. In this section, we also discuss on the semantic interoperability in supply networks and on how we

can use its formal definition to evaluate it. Finally, we describe the role of local ontologies in semantically interoperable systems.

SCOR-FULL ontology identifies and classifies common enterprise notions. However, their semantics is defined externally. We strongly believe that enterprise semantics is well described in many efforts of conceptualizing its architecture, functions and processes and that additional effort in this direction would be redundant. Thus, different enterprise formalizations, contexts and views of existing architectures and other conceptualizations need to be used as sources of specifications of enterprise semantics, and mapped accordingly to the enterprise notions in SCOR-FULL ontology. Currently, SCOR-FULL ontology is mapped to TOVE (Fox et al., 1996) organizational and foundational ontology (in fact, to its OWL representation).

While formal model of supply chain and enterprise semantics provide a theoretical context for semantic interoperability in supply networks, local ontologies introduce actual enterprises contexts, namely, the language which they are going to use to communicate, in a collaboration environment. We believe that enterprises' capability to efficiently collaborate between each other depends on the correspondence between their local semantics and the general context above. Main focus of the work, presented in this paper, is on the analysis of the source of this local semantics, namely relational database systems and, consequently, its explicitation.

The research addresses some of the identified weaknesses of the existing approaches (see section 3.1) to database to ontology mapping and aims at using the OWL expressivity to enrich the implicit semantics of ER (Entity-Relationship) models. It delivers a method and corresponding software tool which: 1) imports the database structure and classifies ER entities; 2) classifies (infers) OWL types and properties; 3) enables lexical refinement and 4) generates local ontology. The concepts of the local ontology are mapped backwards to the corresponding concepts of the intermediary models, in order to enable transformation of semantic to SQL queries. The method and the software tool are described in section 3.2, and are applied in the case of OpenERP database. Some of the experiences gained during implementation of this case are described in section 3.3. The method for execution of semantic queries on the local ontology, namely, instantiation of its concepts according to the content of the relevant database, is described in section 3.4.

It is important to emphasize that the scope of the presented approach is limited only to selected ER patterns which are associated to semantics, expressed by the OWL constructs. Although the process overcomes some of the gaps, identified in the current state-of-the-art in database to ontology mapping, its end result typically requires considerable amount of customization. Since direct mapping is unlikely to produce a useful ontology, the result of this analysis may be considered as intermediary. Thus, it is necessary to put additional work in enactment of this intermediary ontology, which aims at facilitating the final stage of semantic mapping of local ontology to relevant domain ontologies.

2 SEMANTICS OF SCOR AND SEMANTIC INTEROPERABILITY

The concepts and tools presented in this paper are using the formal framework of supply chain

operations presented at Fig. 1. It is developed with goals to enable the semantic interoperability between SCOR-based systems and other relevant enterprise information systems, and to improve the expressivity of SCOR-based models.

In the remainder of this sub-section, we briefly describe the elements of this framework. Their detailed elaboration can be found in Zdravković et al., 2010.

SCOR-FULL ontology is developed by semantic analysis of SCOR Input/Output elements, identification of core terms and their generalization. It extends what we call the SCOR-SYSTEM ontology, which formalizes the SCOR System element. It is then extended by the SCOR-GOAL ontology, which semantically maps its concepts to SCOR Performance Metrics element. SCOR-FULL is exploited by different application models, which formalize specific design goals. For example, SCOR-CFG OWL model is used to develop a semantic web application for supply chain process configuration (Zdravković et al., 2010).

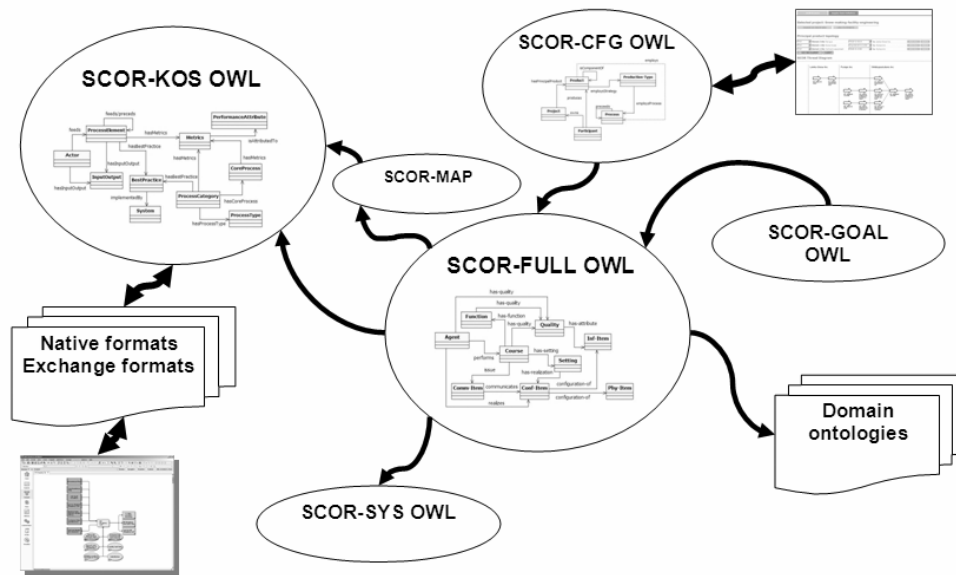


Figure 1: Formal framework of supply chain operations.

The framework is based on a premise that domain knowledge evolves at highest rate at lower levels of abstraction, in domain community interaction. Consensus on the specific notions is more likely to be reached than agreement on the generalizations and abstractions. However, this level is often characterized by the implicit semantics of the standards, reference models, database structures, etc. Thus, we consider coherence between creation,

evolution and use of specific, highly contextualized knowledge and development of formal expressive models as a very important factor for usability of the models.

In the process of development of a formal framework for supply chain operations, we start with modelling the implicit semantics of SCOR model and representing it by using OWL language (SCOR-KOS OWL). OWL (OWL 2 Web Ontology

Language) is a family of knowledge representation languages, which provides the syntax for authoring and exchanging the ontologies among relevant tools and applications. Thus, this common syntax (OWL) enables us to reuse the resulting model with existing tools such as ARIS EasySCOR by IDS or e-SCOR by Gensym. Next, the semantics of the SCOR elements is made explicit through its formalisation with ontology rules embedded into the SCOR-FULL ontology. These rules are used for mapping those concepts to SCOR-KOS OWL concepts. We call this set of rules, SCOR-MAP. SCOR-FULL may then be considered as a micro theory which formalizes knowledge about supply chain operations, by identifying and aggregating common enterprise notions. It is using those concepts to define the semantics of chosen generalizations, namely, the notions of Course, Setting, Quality, Function and Resource.

2.1 Semantic interoperability in supply chain networks

One of the crucial competitiveness factors of the enterprises, especially SME's, is their ability to ensure the quality performance, when simultaneously participating in more than one supply chain, with the same or different products. This ability is mainly driven by the enterprises' capacity to conform to different collaboration (processes) requirements, issued by the customers, where these requirements are described by the different standards or models (SCOR, RosettaNet, ISO/TS 16949 quality specification for automotive industry supply chain and others) and/or methods (CPFR, Vendor-Managed Inventory, etc.).

One of the main challenges is to define the references between the standards, make the corresponding models compatible or complementary and thus, ensure the interoperability of the relevant systems, driven by those models. This challenge can be addressed by formalizing collaboration standards and methods, identifying common enterprise notions (and relating those to the standards' elements) and, mapping them to a general body of knowledge, namely, enterprise models. Hence, the relevant systems, based on those models, will become fully or partially – semantically interoperable.

ISO/IEC 2382 defines interoperability as the “capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units”. Semantic interoperability builds upon this

notion and it means ensuring that the precise meaning of exchanged information is uniquely interpreted by any system not initially developed for the purpose of its interpretation. It enables systems to combine and consequently process received information with other information resources and thus, to improve the expressivity of the underlying ontologies. In our research, we adopt the formal definition of John Sowa (Sowa, 2000; SUO, 2001), because we can use it to evaluate semantic interoperability of enterprise systems:

“A sender's system S is semantically operable with a receiver's system R if and only if the following condition holds for any data p that is transmitted from S to R:

For every statement q that is implied by p on the system S, there is a statement q' on the system R that: (1) is implied by p on the system R, and (2) is logically equivalent to q. The receiver must at least be able to derive a logically equivalent implication for every implication of the sender's system.”

We represent this definition in controlled natural language, as asymmetric logical function semantically-interoperable(S,R):

$$\begin{aligned} & \text{data}(p) \wedge \text{system}(S) \wedge \text{system}(R) \wedge \\ & \text{semantically-interoperable}(S,R) \Rightarrow \\ & \forall p (\\ & \quad (\text{transmitted-from}(p,S) \wedge \\ & \quad \text{transmitted-to}(p,R)) \wedge \\ & \quad \forall q(\text{statement-of}(q,S) \wedge p \Rightarrow q) \\ & \quad \exists q'(\text{statement-of}(q',R) \wedge p \Rightarrow q' \wedge \\ & \quad q' \Leftrightarrow q) \\ &) \end{aligned}$$

Figure 2 illustrates the following assumption of semantic interoperability of systems, represented by the local ontologies: when two different application ontologies of two partners in the supply chain (or two departments or contexts of the same enterprise) are mapped to the same domain ontology, relevant information systems whose knowledge they represent will become fully or partially semantically interoperable in specific direction, depending on the mappings.

In other words, if there exist two isolated enterprise information systems S_1 and S_2 and corresponding application ontologies O_{L1} and O_{L2} and if there are mappings M_{L1D1} and M_{L2D1} , established between the concepts of O_{L1} , O_{L2} and domain ontology O_{D1} , respectively, then there exist mappings M_{L1L2} which can be inferred as logical functions of M_{L1D1} and M_{L2D1} . Each of the local ontologies may (not necessarily) represent one of the contexts of the enterprise (C_1 - C_n).

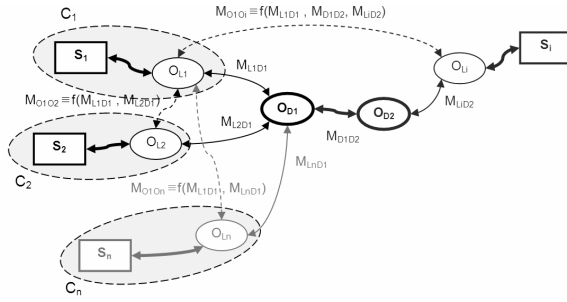


Figure 2: Semantic interoperability of systems.

In our research, we aim at confirming this assumption by inferring the mappings between two contexts of the enterprise, represented by formalizations of an ERP system (S_1 - O_{L1}) and SCOR-based system (S_2 - O_{L2}) for managing the supply chain operations of the enterprise. Mappings are inferred as logical functions of mappings between corresponding formalizations and SCOR-FULL ontology (O_{D1}). Also, by exploiting the mappings between SCOR-FULL (O_{D1}) and TOVE organizational and foundational ontologies (O_{D2}), we aim at showing how the expressivity of the overall ontological environment can be increased, for the benefit of improved semantic interoperability and increased competence. Finally, we evaluate the semantic interoperability of the systems by using the definition above.

In this paper, we focus only on the local ontologies, namely, the formalizations of enterprise information systems' data models and we demonstrate the approach to automated (or semi-automated) generation of local ontologies on basis of the relational database structure.

3 INTEROPERABLE LOCAL ONTOLOGIES

One of the major challenges in the efficient use of computer systems is interoperability between multiple representations of reality (data, processes, etc.) stored inside the systems, or actual representations and reality itself – systems' users and their perception of reality (Hepp, 2007). Where latter can be formalized by the domain ontologies, as shared specifications of the conceptualizations, former relies upon the local ontologies – wrappers for heterogeneous sources of information, business logic and presentation rules.

In our work, the range of semantic interoperability is clearly set to enterprise

information systems. The semantic interoperability of the enterprises is considered as more complex problem and is not addressed in this paper. The conceptualization of their information systems is made on basis of the business logic, which is hidden in the actual code, in most cases, and data model, represented by the corresponding relational database structure. We consider EIS's databases as legitimate starting point for building a relevant local ontology. Obviously, business logic which is encapsulated in the EIS' will remain hidden – only underlying data model is exposed by ontology. The exceptions are database's triggers, which can be considered as business rules, if they are not implemented only to enforce referential integrity of the database.

In the remainder of this section, some of the reported work in database to ontology mapping is presented. Then, our approach to local ontology generation is described and demonstrated on the case of ER model (database) of OpenERP enterprise software.

3.1 State of the art in database to ontology mapping

Review of the relevant literature reveals several approaches which address database to ontology mapping. In this section, we present the main features of four distinctive frameworks, made with different objectives, and we identify gaps, in terms of the usability and coverage of the frameworks.

Work on DB2OWL mapping facility is a part of development of a general interoperability architecture (Ghawi and Cullot, 2007) that uses ontologies for explicit description of the semantics of information sources, and web services to facilitate the communication between the different components of the architecture. DB2OWL (Cullot et al., 2007) looks for some particular cases of database tables to determine which ontology component has to be created from which database component. According to these cases, conversion process is performed (table \rightarrow class, column \rightarrow property, constraint \rightarrow relation) where the set of correspondences between database and ontology components is conserved, thus enabling the translation of ontological to SQL queries and retrieval of corresponding entities. However, it remains unclear how this translation will be implemented. More important, the semantics of existential constraints of the columns and cardinality of relations is not taken into account.

Relational.OWL (de Laborda and Conrad, 2005) is a candidate for data and schema representation

format, relevant for database to ontology mapping. It provides a meta-model, which describes the components of the relational database. Hence, it can be used as an intermediary in the process of database to ontology mapping, instead of a document with correspondences, used by DB2OWL. Unfortunately, it suffers from the same problems as DB2OWL - multiplicity of the foreign keys is not in the model. Thus, it is not possible to use it to assign source and destination cardinality to OWL properties. Moreover, source multiplicity determines important aspect of the semantics of the underlying concept or database table. Namely, where source multiplicity of the foreign key is 1, the corresponding OWL relation shall be necessary condition for instantiation of the concept in its domain. This is important semantic feature, because it enables intensional conceptualization of the entity.

Where DB2OWL is used to create new ontology from existing schema, D2OMapper (Xu et al., 2006) is a tool for automatic or semi-automatic creation of the mappings between database schema and existing ontology. D2OMapper follows a set of predefined heuristic rules, based on the conceptual correspondences between the schema and ontology. This work is based on the authors' experience in developing ER2WO (Xu et al., 2004) tool for translating ER schema into OWL ontology.

Vis-A-Vis tool (Konstantinou et al., 2006) uses the Protégé libraries for graphically representing an ontology, a database model (MySQL or PostgreSQL) and the mappings between them. The

plug-in allows queries to be asked to the ontology and returns results from the database. The key motivation of the authors was to keep the instances stored in a database and maintain a link to the dataset. Thus, ontologies become smaller.

3.2 Our approach to database to ontology mapping

Mapping is a process in which implicit semantics of a database schema is mapped to the explicit and formal knowledge structure of the ontology. In our approach, we use the database schema to generate this formal structure, while preserving the logical mappings between ER meta-model and generated local ontology. These mappings will enable the translation of semantic to database queries.

Generation process consists of 4 phases: a) data import and classification of ER entities; b) classification (inference) of OWL types and properties; c) lexical refinement; d) generation of local ontology; and is illustrated on Figure 3, below. The process is supported by a web application, developed by using RAP API (Oldakowski et al., 2005), a PHP-based package for parsing, querying, manipulating, serializing and serving RDF models. Web application consists of modules for data import/assertion of ER meta-model instances, lexical refinement and transformation of classified OWL types and properties to a local ontology.

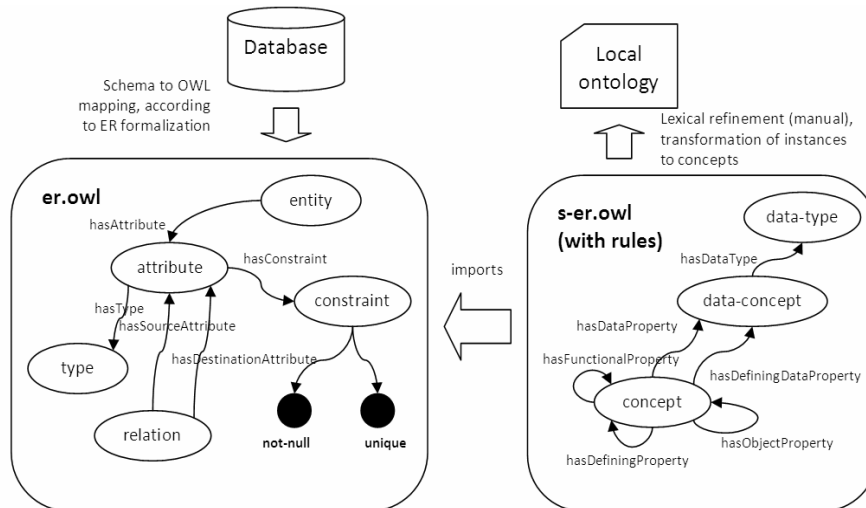


Figure 3: Approach to database-to-ontology mapping.

First, database schema is investigated and OWL representation of the ER-model is constructed. This is realized by developed application, which connects

to the database, uses introspection queries to discover its structure and asserts the relations between the artifacts by using proposed ER

formalization (er.owl). Following assertions are made for each field of the corresponding table: hasAttribute (entity, attribute), hasType(attribute, type) and hasConstraint(attribute, 'not-null') and/or hasConstraint(attribute, 'unique') (if applicable). Following assertions are made for each relation: hasDestinationAttribute (relation, attribute), hasSourceAttribute(relation, attribute).

Second, resulting (serialized) OWL representation of the database ER-model is imported into meta-model (s-er.owl), which classifies future OWL concepts (Ax1) and domains and ranges of the object and data properties, according to defined axioms (Ax2, Ax4). Although specification of object and data properties may impose the unnecessary restrictions on the resulting ontology, we consider those as important for improving the efficiency of mapping or alignment process, which is critical for the semantic interoperability. Another reason for the assertion of object properties in OWL representation of database ER-model is that object properties of the resulting local ontology will be annotated with the URI's of the respective relations, in order to enable the correspondence between the ontology and database representation, for the benefit of query transformation. On the other hand, existential constraints from the ER-model are associated to an explicit semantics of the resulting ontology, namely, necessary conditions for entailment of the corresponding concepts. According to these constraints, axioms for intensional conceptualization (necessary conditions, or inherited anonymous classes) for particular entity are identified by inferring ranges of hasDefiningProperty(concept, concept) and hasDefiningDataProperty(concept, data-concept) relations (Ax2.2 and Ax4.2). Finally, the approach takes into account the functionality of the properties (owl:FunctionalProperty). Functional property is property that can have only one (unique) value y for each instance x. They are classified when relation one-to-one is identified between two concepts (Ax2.3).

Classification of future OWL concepts is inferred by exploiting following axioms:

Ax1. Concepts are all entities of the ER model's OWL representation, except the entities whose all attributes are relation sources (corresponding to intermediary tables, connecting two tables with many-to-many relationship).

$$\text{er:entity}(x) \wedge \text{not } (\text{er:hasAttribute only } (\text{er:attribute } \wedge (\text{er:isSourceAttributeOf some } \text{er:relation}))) \Rightarrow \text{s-er:concept}(x)$$

Ax2.1. Domains and ranges of the object properties are inferred by using the rule below.

$$\begin{aligned} &\text{er:entity}(x) \wedge \text{er:entity}(y) \wedge \\ &\text{er:relation}(r) \wedge \text{er:hasAttribute}(x, a1) \wedge \text{er:hasAttribute}(y, a2) \wedge \\ &\text{er:isDestinationAttributeOf}(a2, r) \wedge \\ &\text{er:isSourceAttributeOf}(a1, r) \Rightarrow \text{s-er:hasObjectProperty}(x, y) \end{aligned}$$

Ax2.2. Domains and ranges of the defining properties (necessary conditions of the concept) are inferred by using the rule below. Defining property is a sub-property (rdfs:subPropertyOf) of the object property (hence, simplified representation of the rule below).

$$\begin{aligned} &\text{s-er:hasObjectProperty}(x, y) \wedge \\ &\text{er:hasConstraint}(a1, \text{'not-null'}) \Rightarrow \text{s-er:hasDefiningProperty}(x, y) \end{aligned}$$

Ax2.3. Domains and ranges of the functional properties are inferred by using the rule below. Functional property is a sub-property (rdfs:subPropertyOf) of the defining property (hence, simplified representation of the rule below).

$$\begin{aligned} &\text{s-er:hasObjectProperty}(x, y) \wedge \\ &\text{er:hasConstraint}(a1, \text{'not-null'}) \Rightarrow \text{s-er:hasDefiningProperty}(x, y) \end{aligned}$$

Ax3. Data concepts are all attributes of the ER model's OWL representation which are not at the source of any relation.

$$\begin{aligned} &\text{er:attribute and not} \\ &(\text{er:isSourceAttributeOf some } \text{er:relation}) \Rightarrow \text{s-er:data-concept} \end{aligned}$$

Ax4.1. Domains and ranges of the data properties are inferred by using the rule below. Ranges of the data properties are data types, corresponding to the simple types from XML schema.

$$\begin{aligned} &\text{er:type}(x) \Rightarrow \text{s-er:data-type}(x) \\ &\text{s-er:concept}(c) \wedge \text{er:attribute}(a) \wedge \\ &\text{er:type}(t) \wedge \text{er:hasAttribute}(c, a) \wedge \\ &\text{er:hasType}(a, t) \Rightarrow \text{s-er:hasDataProperty}(c, t) \end{aligned}$$

Ax4.2. Domains and ranges of the defining data properties are inferred by using the rule below. Defining data property is a sub-property (rdfs:subPropertyOf) of the data property (hence, simplified representation of the rule below).

$$\begin{aligned} &\text{s-er:hasDataProperty}(c, t) \wedge \\ &\text{er:hasConstraint}(a, \text{'not-null'}) \wedge \\ &\text{er:hasConstraint}(a, \text{'unique'}) \Rightarrow \text{s-er:hasDefiningDataProperty}(c, t) \end{aligned}$$

Rules above classify instances of the OWL representation of the database ER model (er.owl) into a meta-model (s-er.owl). Inferred triples can be edited in a simple web application, which also launches the process of local ontology generation. In

3.3 Case implementation

With all modules installed, OpenERP database counts 238 tables. In the first step of database import into er.owl model, namely, instantiation of the OWL representation of ER model, 3806 individuals are created (2633 attributes, 238 entities, 934 relations) and 7999 object property assertions are made. In the second step of classification of OWL concepts and properties, 696 of individuals' entailments (193 concepts and 493 data-concepts) and 2779 properties are inferred, on the basis of axioms, presented in section 3.2. All inferences are stored in a separate OWL file in order to reduce the processing requirements for the final step. In the final step of local ontology generation, application transforms classified instances of the meta-model of the openERP database to the corresponding OWL concepts and properties (see Fig. 4).



semantic query executed against the local ontology O_{Li} would normally infer triples of information from the database of S_i . However, if mappings (or logical function of mappings) between O_{Li} and O_{Lj} exist, inferred triples will also include information from the database of S_j . For example, in supply chain networks, a single semantic query can be used to find out the availability of specific resource or competence, of all - owned and used by the enterprises from the Virtual Breeding Environment (for the benefit of virtual enterprise formation process).

3.4 Reasoning with local ontologies and translation of semantic to SQL queries

Semantic query can be considered as a pair (O, C), where O is a set of concepts which need to be inferred and C - a set of restrictions to be applied on their properties, namely value (owl:hasValue and qualified cardinality restrictions, owl:allValuesFrom, owl:someValuesFrom) and cardinality constraints (owl:cardinality, owl:minCardinality, owl:maxCardinality). This consideration corresponds to a simplified representation of a SQL query which includes tables (and fields) and comparison predicate, namely restrictions posed on the rows returned by a query. In addition, different types of property restrictions correspond to different cases (or patterns, where complex semantic query is mapped) of SQL queries.

Where relevant entailments can be reasoned only by property domain and range inferences, a set C may be considered as necessary and sufficient for representation of the semantic query. For example, in openERP ontology (see Fig. 4), a DL query “hasAccountAccountType some (hasCode value 3)” returns all instances of account_account concept whose type’s code is exactly 3. This kind of query representation (only by using properties restrictions) may produce unpredictable and misleading results where the restrictions are posed on the common lexical notions of different concepts, such as “name”, “type”, “id”, etc. Ambiguity of the corresponding properties is reflected on the relevant ontology in the sense that their domains are typically defined as union of large number of concepts. For

example, in openERP ontology, domain of the “hasName” data property is union of 170 concepts. However, this ambiguity may be considered as an advantage in some cases. Value restrictions on ambiguous data properties may produce relevant inferences and thus, facilitate semantic querying without a need to have extensive knowledge on the underlying ontology structure. This kind of query is mapped to a SQL UNION query which combines SELECT sub-queries made on the each element of the property domain, with the WHERE statement corresponding to the relevant rows restrictions. For example, in a mapping process, DL query “hasName value ‘Derek Porter’” is first used to infer all 170 possible entailments (property domains), which are, then, used to assemble qualified (O,C) pairs, e.g. “res_users and hasName value ‘Derek Porter’”. When corresponding element of the UNION query is assembled, a static field with appropriate label (a reference to the concept) is added to each of the elements, so as to become possible to decide on the entailments. In other words, we need this to determine which sub-query actually returned the results.

In the first step of the method, decomposition and semantic analysis of the input query is performed. The 4-tuplets in forms of (subject predicate some|only|min n|max m|exactly o bNode) and (subject predicate value {type}) are extracted from the input query. In case of the DL query which returns all concepts which are related to a company whose primary currency is EURO (“hasResCompany some (hasResCurrency some (hasName value "EUR"))”), following 4-tuplets are identified:

```
X hasResCompany some bNode1
bNode1 hasResCurrency some bNode2
bNode2 hasName value "EUR"
```

Next, a database connection is established and SQL query is constructed and executed for each 4-tuplet, in reverse order, as a result of analysis described above. Each query returns data which is used to generate OWL statements which are asserted to a temporary model. Each set of the OWL statements corresponds to a sub-graph whose focal individual is an instance of the concept, inferred on basis of the 4-tuplet’s property domain or returned result (label). Other individuals or values correspond to defining properties of this concept (inherited anonymous classes). In case of ambiguity, resulting blank nodes are represented as the sets, which are filtered as a result of range inference of the parent 4-tuplet, in a final stage of the method.

4 CONCLUSIONS AND FUTURE WORK

Work, presented in this paper is a part of the research of semantic interoperability in supply chain networks. This research is based on formalization of widely adopted supply chain process reference model and includes development of its OWL representation, semantically enriched model, specification of some of its entities (namely, system and goal) and correspondences with other models. It transforms implicit semantics of the reference model to the explicit specification which uses common enterprise notions, assumingly defined in other domain ontologies and/or conceptualizations of relevant enterprise models, architectures and frameworks. Used approach is characterized by the multiple, cross-referenced levels of abstraction, represented by the OWL models of different expressivity. Modular design contributes to the usability of the ontology framework, by avoiding performance related problems in reasoning, as well as by providing increased potential for ontology matching. Thus, it is expected to facilitate the semantic interoperability in supply chain networks.

In this paper, we focus on introducing the partial realities of the enterprises, namely data representations of their information systems, into heterogeneous environment of a supply chain network. In presented approach, enterprise data models are used to generate local ontologies, by applying a set of rules for interpreting the semantics of an ER model, namely database schema. Although “database to ontology mapping” is not a novel concept, we show that existing approaches are characterized by weaknesses, most of which are related to lack of completeness of properties’ semantics. Our approach and corresponding tools aim at overcoming those, thus enabling the complete (from the aspect of OWL expressivity) interpretation (explicitation) of the implicit semantics of the ER model, as well as full correspondence between semantic and database queries.

In context of the semantic interoperability in supply chain networks, resulting local ontologies may be considered as enterprise message models. As such, they aim at enabling the semantic interoperability of corresponding enterprise information systems, not the enterprises themselves. Still, significant research efforts are needed for representation and exposition of the enterprise business logic, which is hard-coded in the systems, as well as the semantics of the instances, namely information which is stored in the database (for

example, occurrence patterns). Another line of research in the future will aim at enactment of the generated ontologies, as they are considered only as intermediary models. We consider those research directions as important for increasing collaboration in a supply chain network, as its fulfilment will enable logic driven, automatic and transparent decision making, thus, facilitating a transition from traditional supply chains to virtual enterprise and related paradigms.

ACKNOWLEDGEMENT

The work presented in this paper was partially supported by the program for scientific cooperation between Serbia and France PHC PAVLE SAVIC, project N° 23494VF (2010/2011).

REFERENCES

- Browne, J. and Zhang, J. (1999). Extended and virtual enterprises – similarities and differences. *International Journal of Agile Management Systems*, 1(1), 30-36.
- Cullot, N., Ghawi, R. and Yetongnon, K. (2007). DB2OWL: A Tool for Automatic Database-to-Ontology mapping. In: Proceedings of the 15th Italian Symposium on Advanced Database Systems. June, 17-20. 2007, Torre Canne di Fasano (BR), Italy.
- de Laborda, C.P. and Conrad, S. (2005). Relational.OWL: a data and schema representation format based on OWL. In: Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling. January 30 - February 4, 2005. Newcastle, Australia.
- Fox, M.S., Barbucaanu, M. and Gruninger, M. (1996). An organization ontology for enterprise modeling: preliminary concepts for linking structure and behavior. *Computers in Industry*. 29 (1-2) 123-134.
- Ghawi, R. and Cullot, N. (2007). Database-to-Ontology Mapping Generation for Semantic Interoperability. In: Proceedings of 3rd International Workshop on Database Interoperability. September, 24. 2007, Vienna, Austria.
- Hepp, M. (2007). Ontologies: State of the art, business potential and grand challenges. In Hepp, M., De Leenheer, P., de Moor, A. and Sure, Y. (eds), *Ontology Management – Semantic Web, Semantic Web Services and Business Applications* (pp.3-22), Berlin/Heidelberg: Springer.
- Jespersen, B.D. and Larse, T.S. (2006). *Supply Chain Management - in Theory and Practice*. Copenhagen:Business School Press.
- Konstantinou, N., Spanos, D., Chalas, M., Solidakis, E. and Mitrou, N. (2006). VisAVis: An approach to an intermediate layer between ontologies and relational database contents. In: Proceedings of International Workshop on Web Information Systems Modeling. June, 6, 2006, Luxembourg.
- Lamber, D.M. and Knemeyer, A.M. (2004, December). We're in This Together. *Harvard Business Review on Supply Chain Management*.
- Obrst, L. (2003). Ontologies for semantically interoperable systems. In: Proceedings of the 12th International Conference on Information and Knowledge Management. November 3-8, 2003, New Orleans, USA.
- Oldakowski, R., Bizer, C. and Westphal, D. (2005). RAP: RDF API for PHP. In: Proceedings of the 1st Workshop on Scripting for the Semantic Web, May 30, 2005. Heraklion, Greece.
- OWL 2 Web Ontology Language (2009). Document Overview. W3C Recommendation 27 October 2009. <http://www.w3.org/TR/owl2-overview/>
- Sánchez, N.G., Apolinar, D., Zubiaga, G., Atahualpa, J., González, I. and Molina, A. (2005). Virtual Breeding Environment: A First Approach to Understanding Working and Sharing Principles. In: Proceedings of the 1st International Conference on interoperability of Enterprise Software and Applications. February 23-25, 2005, Geneva, Switzerland.
- Sowa, J. (2000). Knowledge Representation : Logical, Philosophical, and Computational Foundations, CA:Brooks/Cole Publishing Co.
- Stewart, G. 1997. Supply-chain operations reference model (SCOR): the first cross-industry framework for integrated supply-chain management. *Logistics Information Management*. 10 (2) 62-67.
- SUO (2001). The IEEE Standard Upper Ontology web site. <http://suo.ieee.org>
- Williamson, O. (1985). *The Economic Institutions of Capitalism: Firms, Markets, Relational Contracting*. New York:The Free Press.
- Xu, Z., Zhang, S. and Dong, Y. (2006). Mapping between Relational Database Schema and OWL Ontology for Deep Annotation. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. 18-22. December, 2006, Hong Kong.
- Xu, Z., Cao, X., Dong, Y. and Su, W. (2004) Formal Approach and Automated Tool for Translating ER Schemata into OWL Ontologies. *Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science*, Vol. 3056/2004, 464-475.
- Zdravković, M., Panetto, H. and Trajanović, M. (2010). Towards an approach for formalizing the supply chain operations. In: Proceedings of the 6th International Conference on Semantic Systems. September, 6-8, 2010, Graz, Austria.