

# A Semantic Solution for Seamless Data Exchange in Supply Networks

Elena Andreeva, Tatiana Poletaeva, Habib Abdulrab, and Eduard Babkin

**Abstract** This paper proposes the semantic solution for data exchange in dynamically changing supply networks. Because of the dynamic nature of supply networks, there is a necessity to manage their efficiency. The first step in this direction is proper data exchange, which leads to transparency of networks and speeds up interactions of their participants. Despite plenty of data standards, there is a lack of approaches to data modeling allowed the seamless knowledge sharing within supply networks. This paper introduces a new ontology-based data metamodel of supply networks based on organizational ontology, consistent theory for data modeling, and the ontologized standard in logistics domain. At the first part of the paper, selected approaches to data modeling are explained and justified. Then, basic information patterns applicable for modeling of supply networks are explained. Finally, we elaborate how the proposed ontological framework facilitates knowledge integration in logistics.

**Keywords** Semantic data modeling • Domain ontology • Knowledge management • Semantic interoperability • Supply network

## 1 Introduction

Nowadays the structure of supply networks is getting more complex with the increasing volume of transportation load, the number of people and organizations involved in logistical processes, stronger requirements to the quality of product distribution, and the growing volume of information flow. Therefore, businesses realize the necessity of applying information systems to facilitate the control of products within supply networks. However, seamless information exchange between

---

E. Andreeva (✉) • T. Poletaeva • E. Babkin  
National Research University Higher School of Economics, Nizhny Novgorod, Russia  
e-mail: [eandreeva@hse.ru](mailto:eandreeva@hse.ru); [ta.poletaeva@gmail.com](mailto:ta.poletaeva@gmail.com); [eababkin@hse.ru](mailto:eababkin@hse.ru)

H. Abdulrab  
INSA de Rouen, LITIS Lab., Rouen, France  
e-mail: [abdulrab@insa-rouen.fr](mailto:abdulrab@insa-rouen.fr)

information systems of supply network parties is hindered by poor semantic interoperability, in general, and data integration problems, in particular [1].

The lack of semantic interoperability of information systems comes out from the impossibility to declare the universal standard for naming and meaning of measures and characteristics of the supply network among its current and potential future parties. The storage and transmission of logistics records have been the object of many initiatives concerning standardization, to wit: Universal Business Language, UBL (OASIS project),<sup>1</sup> UN/CEFACT<sup>2</sup> logistics module, GS1<sup>3</sup> Logistics Interoperability Model, etc. Nevertheless, these standards are not amenable to foster semantic interoperability, because their metamodels are not smoothly integrable. In addition to the use of different standards and approaches to data modeling, developers of information systems specify data storages with the particular concepts and data patterns, which are rather based on their experience and intuition than on any theoretical foundations [2]. The following commonly encountered problem is a shining example of the problems of information exchange. The attribute “selling price” of the same products has different meanings and, therefore, different values for manufacturers and distributors. For a manufacturer the selling price of a product equals the sum of net cost and extra cost. But a distributor considers the selling price to be the sum of purchase price and some extra costs. As for other cases, the root of heterogeneity lies in the differences of the definitions given for the same concept. Aforementioned lack of semantic interoperability occurs when automated discovery of concept definition and meaning in a data model is not possible.

The widely known solution for the problem of information exchange is the use of domain ontologies. Logistics domain ontologies provide a theory to address semantic interoperability and data/standard integration between information systems of supply network parties, allow automatic analysis by means of artificial intelligence, and convey a knowledge repository. As soon as the elements of particular data models are mapped onto complete collection of logically related concepts provided by the domain ontology, the problem of information exchange is resolved. Meanwhile the mapping between a particular data model and the ontology-related data patterns is easier than the mapping between two particular data models, because the meaning of ontological data elements is unambiguously defined by their interrelations and related axioms [3]. Creation of underlying domain ontology is propagated by many standards in different domains (ISO 15926,<sup>4</sup> IDEAS,<sup>5</sup> DoDAF,<sup>6</sup> OMG FIBO,<sup>7</sup> etc.). However, there are only some attempts of creation the domain ontology in logistics [4–6].

---

<sup>1</sup>OASIS: <https://www.oasis-open.org/>.

<sup>2</sup>UN/CEFACT: <http://www.uncece.org/CEFACT/>.

<sup>3</sup>GS1: <http://www.gs1.org/>.

<sup>4</sup>ISO 15926: <http://15926.org/>.

<sup>5</sup>IDEAS: <http://ideasgroup.org/>.

<sup>6</sup>DoDAF: <http://dodcio.defense.gov/TodayinCIO/DoDAArchitectureFramework.aspx>.

<sup>7</sup>OMG FIBO: <http://www.omg.org/>.

The main drawback of known approaches in building the logistics domain ontologies is the underestimation of the importance of the underlying foundational ontology. Prevailing Entity paradigm [2] for data modeling leaves high levels of freedom in the definition of entities and their attributes. On the other hand, it provides a very weak theory about parts and wholes, types and instances, identity, dependency, and unity. That is why relational data models contain ambiguous definitions of real objects. Consequently, it is difficult to automate the mapping process and the integration of the data models based on Entity paradigm.

Our research is aimed to overcome the problem of information exchange in logistic networks by developing a logistic domain ontology based on the Object paradigm by means of the business objects reference ontology (BORO) methodology [2]. In opposition to the Entity paradigm, the Object paradigm of data modeling [2] has no superficial division of the world into entities and attributes. Everything in the world is considered as an explicit map of objects and their patterns of relationships [2], whether it is individual, class, individuals' relationship, or class of relationships. The Object paradigm, fully expressed in the BORO methodology [2], (1) provides strong logical foundations for classification and identity; (2) provides clearer and more precise representation of reality based on spatial and temporal dimensions of objects; (3) provides the ability to capture dynamic objects showing their changes over time; and (4) allows avoiding subjective and possibly erroneous ontological assumptions in modeling constructs [7]. That is why we chose this paradigm for modeling of dynamic supply networks. BORO-based formal ontology has been already designed by the IDEAS Group. IDEAS ontology was used for developing of data-modeling standards—DoDAF, MODAF, and MODEM. Despite the fact that the IDEAS-based data standards were developed to support the exchange and sharing of military enterprise architectures, IDEAS upper-level ontology is domain-independent [8]. In our work we partially use the IDEAS upper-level ontology, and extend it with the concepts related to physical flows of enterprises.

In order to build the domain-specific level of the logistics ontology, we need a comprehensive set of concepts that are used in this field. For this purpose we chose supply chain operations reference model (SCOR model) [9] as a standardized description of business operations in supply chains and the source of unambiguously defined terms that participants of supply networks operate.

Thus, in this paper we present the data metamodel built in accordance with the BORO methodology partially using the IDEAS upper-level ontology. Moreover, the proposed metamodel is built on top of the formal enterprise ontology described in detail in [10]—a core ontology for supply networks. Providing essential ontologized domain-independent knowledge about enterprises, this ontology forms a shape for the integration of heterogeneous data models in the logistics domain. The domain-specific level of the metamodel is built by applying business objects of the SCOR model to the domain-independent level. The proposed metamodel is aimed to facilitate significantly the supply chain management and simplify interaction and data exchange.

The outline of the paper is as follows. First, the theoretical background of proposed ontology is summarized in Sect. 2. Proposed logistics ontology and basic

information patterns applicable for modeling of supply networks are explained in Sect. 3. Then, in order to verify the completeness and coherence of the developed ontology, we used the ontology to build the data metamodel of a particular supply process. The details of this data metamodel are described in Sect. 4. In Sect. 5 we demonstrate the semantic power of the metamodel by the knowledge that can be extracted from related data model. Moreover, in this section we elaborate on the integration of the particular logistics data metamodel with the proposed one. Finally, Sect. 6 provides conclusions and directions for further research.

## 2 The Ontological Engineering Approach

In our research the BORO methodology [2] is a theoretical basis for the ontology development. The Object paradigm of data modeling underlies the BORO methodology and defines some fundamental approaches to data modeling.

Firstly, according to the BORO, only the concepts of tangible objects that exist in a real world should form the ontological core. Moreover, object identity is defined as an object's extension in the universe [7]. The extensional approach provides an ontology that is well suited to deconflicting multiple identification schemes [11]. The second modeling principle is the consideration of spatio-temporal (4D) extensions of any object. This approach intrinsically represents the temporality of objects by the assumption that they just partly exist at each time instant of their life span. Together with the essential information patterns listed hereafter, this modeling principle makes the methodology perfectly suitable for designing data models of dynamically changing logistic networks.

The ontological constructs used in the Object paradigm are individuals, classes, and tuples (coupled relations) [2]. It is assumed that all these constructs represent four-dimensional objects. Individuals are four-dimensional objects that persist through time. A class is an object according to its definition as a collection of 4D-individuals. A tuple as the ordered pair of interconnected individuals or classes is also an object. Any object is a sequence of 4D-states within its lifecycle. Any state is bounded by the events that mark its beginning and the end. Since an event happens at a point of time, it has no time duration, but, in turn, it has a spatial extension. A spatial extension of an event is a time slice that is a 3D-part of an object.

Finally, in the Object paradigm C. Partridge proposed fundamental information patterns for expressing relationships between 4D-objects. He proposed using the "whole-part" pattern to assign that an object is a part of another one. Also the sequence of two states of an object can be related by means of the "before-after" pattern. The role of the "pre-condition" pattern is to link an event with the conditions required to make it happen.

The Object paradigm proposes the unique description of production processes as the set of four-dimensional objects involved into this process. Consequently, a process is also considered as a 4D-object. Thus, the extension of a simple process can be the aggregation of the extensions of an actor and a product changed by the

actor. Changes of the states of constituent objects define the states of the process. The sequence of process states is unambiguously expressed by the means of the “before-after” and “pre-condition” patterns.

In accordance with the Object paradigm, the BORO methodology allows revealing business objects as well as relationships between them. Moreover, the methodology defines some conceptual modeling rules for defining new ontological concepts.

### 3 The Logistics Ontology

In this section we explain the developed ontology and we give the notion of invented concepts and the reasons why they were added to the ontology.

The developed ontology consists of three levels of abstraction: the Framework, the Application level, and the Operational level [2]. Because of the domain requirements, the conceptualization of physical flows of enterprises (the Application level) was created upon the formal enterprise ontology (the Framework) described in detail in [10]. The Application level contains the concepts of standard supply processes specified by the SCOR model. This ontological level is supposed to be the common core of different data metamodels of supply networks putting aside their specifics. The Operational level in turn comprises business objects related to a particular business scenario. Also, as it was mentioned in the Introduction, the notation of the IDEAS standard [8] was used for the visualization of created conceptual model.

The top level of the Framework fully corresponds to the BORO fundamental ontology. According to the extensional approach, all concepts have corresponding

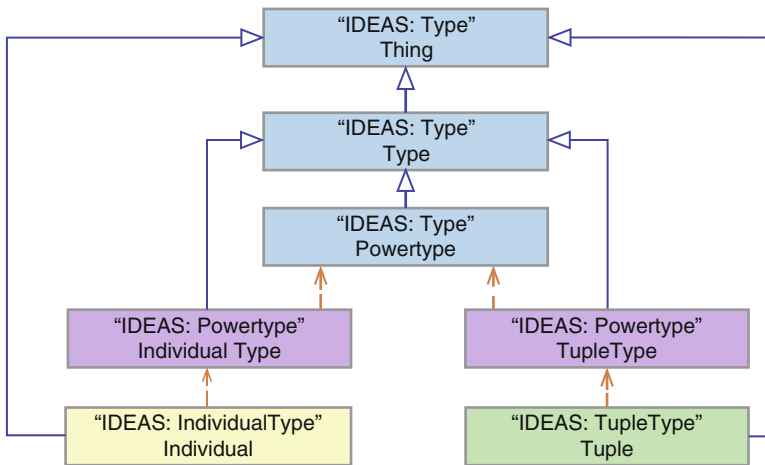


Fig. 1 The top concepts of the Framework

spatial equivalents. Thus, the top concept presented in Fig. 1 is a *Thing*—“a union of *Individual*, *Type*, and *Tuple*” [2] where an *Individual* is an object that has a spatio-temporal extent, i.e. it is tangible in our world [2], a *Tuple* is “a relationship between two or more things” [2]. Spatio-temporal extension of a tuple comprises the extensions of its placeholders. Specified set of individuals or tuples is a *Type* (class). According to the IDEAS notation, yellow rectangles depict classes of *Individual* objects, green rectangles—classes of relationships, and purple rectangles—classes of classes of *Individual* or *Tuple* objects in figures of this section. A type and its members are connected through *classification* relationships (presented by dotted brown lines in the graphical notation of the metamodel). A relationship between a type and one of its subtypes is the *specialization* relationship (presented by navy blue arrow in the graphical notation of the metamodel).

Specialization of the *Type* depends on the nature of its possible members. Thus, the *IndividualType* is the set of classes with members, which are *Individual* instances. The *TupleType* is the set of classes with members, which are *Tuple* instances. A class that has other classes as its members is called *Powertype* (a class of classes). Subclasses of the *Type* can relate to each other. Thus, the *IndividualType* and the *TupleType* are instances of the *Powertype* class.

Since the model reflects the concepts of the enterprise ontology [12], it includes several categories of objects: agents, roles, resources, processes, transactions, and facts. The concept *Agent* represents people or organizations that are authorized and can take the responsibility to participate in business processes. *Resource* concept expresses any kind of resources involved into processes, and *Product* as a subclass of *Resource* is a class of end products. *Process* is an object of *Individual* type whose extent is defined by its involvements [2]. And finally, world changes (facts) and agents’ interactions about these facts (transactions) are represented in the metamodel in accordance with DEMO enterprise ontology [12] by *Transaction*, *CoordinationFact*, and *ProductionFact* concepts. Speaking in the language of BORO we can consider facts as 3D-objects, events that change the state of business processes. Besides, the data model includes classes of relationships between given concepts. These relationships allow linking objects with their parts and their states; they allow assigning process states and transactions into one sequence, relating processes and involved roles, agents, and resources.

Following BORO fundamental ontology, proposed conceptual model exploits four fundamental information patterns: whole-part, before-after, pre-condition, and participation. For assigning agents, resources, and processes with their parts and states the following concepts are used: *agentWholePart* and *agentWholeState*, *resourceWholePart* and *resourceWholeState* (Fig. 2), and *processWholePart* and *processWholeState*.

To link two successive states of an object BORO provides with *beforeAfter* pattern. The *beforeAfter* tuple class has two subclasses: *resourceStateBeforeAfter* and *processStateBeforeAfter* (Fig. 3). This specialization will be needed in further computer processing of the data model.

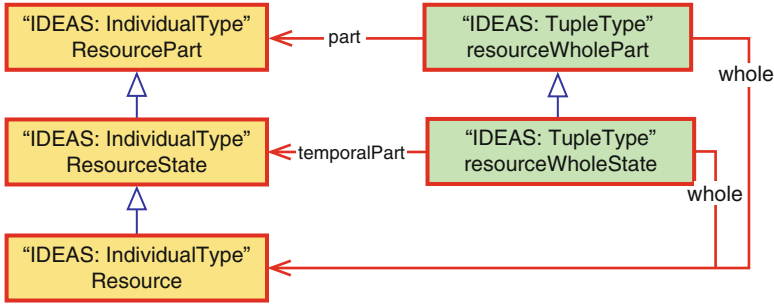


Fig. 2 resourceWholePart and resourceWholeState patterns

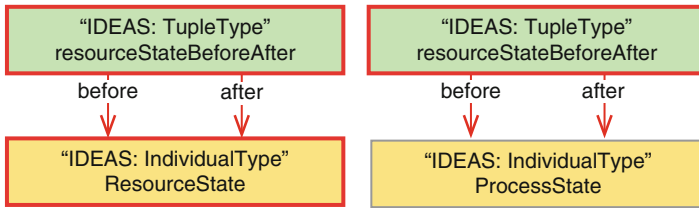


Fig. 3 beforeAfter pattern

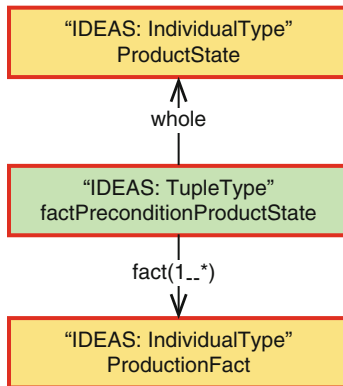


Fig. 4 factPreconditionProductState pattern

Links between events and their preconditions are important in the domain of supply networks. For that reason *factPreconditionProductState* concept was added to the metamodel (Fig. 4). In human language it means that a resource will not take a certain state unless a fact happens.

In the domain of supply networks a process is performed by some actors and involves some resources or products. This kind of relations is supported by means of active (*agentParticipation*) roles (Fig. 5) and passive (*passiveParticipation*) (Fig. 6). When Agent takes a certain role in a process, it becomes an *ActiveParticipationExtent* related to this process (Fig. 7).

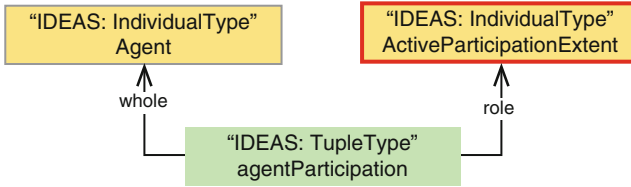


Fig. 5 *agentParticipation* pattern

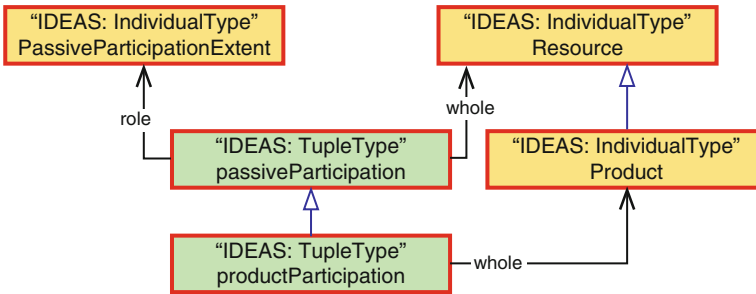


Fig. 6 *passiveParticipation* pattern

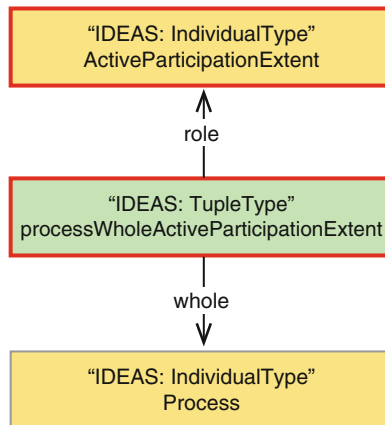


Fig. 7 *processWholeActiveParticipationExtent* pattern

But the notion “role” does not define responsible actors for certain processes. To do this we use *responsibleForProductionFact* class of relationships (Fig. 8). The responsibility for coordination facts is expressed by *responsibleForCoordinationFact* pattern.

Following the enterprise ontology [12], we consider a process as a sequence of transactions. The model reflects essential parts of any transaction: coordination and production facts. A process is assigned with transactions through



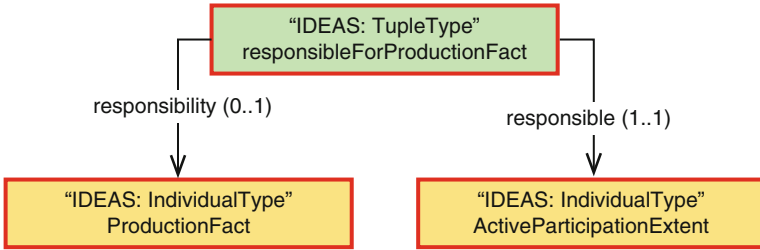


Fig. 8 responsibleForProductionFact pattern

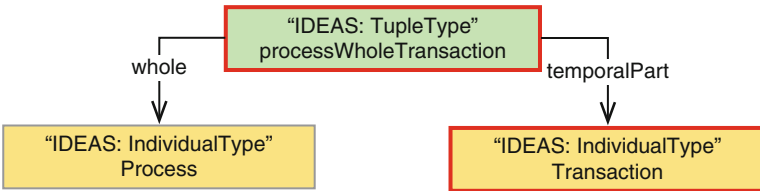


Fig. 9 processWholeTransaction pattern

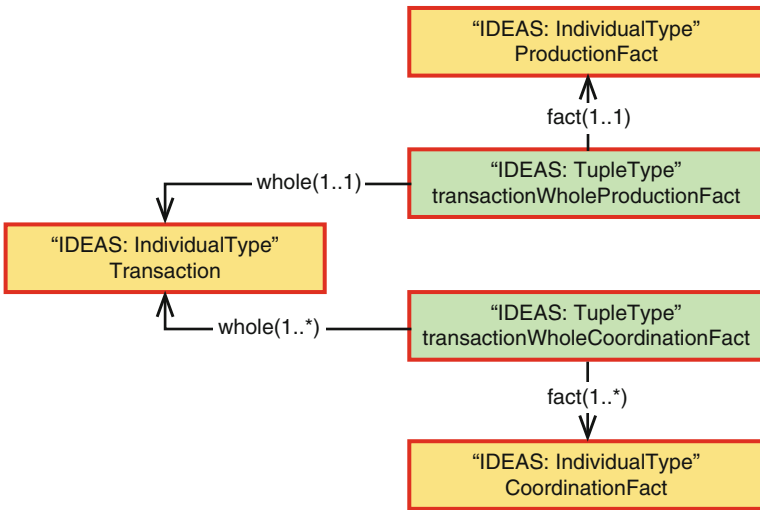


Fig. 10 transactionWholeProductionFact and transactionWholeCoordinationFact patterns

processWholeTransaction pattern, and a transaction is assigned with facts through transactionWholeFact pattern. Relevant diagrams are shown on Figs. 9 and 10.

The Application level is the specification and extension of the Framework in application to logistics domain. The concepts of this level express specific kinds of processes, transactions, resource states, and roles that may appear in supply networks according to the supply chain operation reference (SCOR) model [9].

To that moment Source Stocked Resource and Deliver Stocked Resource processes [9] were translated from the standard to the ontology. By analyzing the description of these processes given in the SCOR standard there were revealed ontological transactions, production, and coordination facts, actor roles, and relationships between all of these classes of objects. The process *SourceStockedResource* includes five transactions: *SchedulingDeliveries*, *Receiving*, *Verification*, *Transferring*, and *Payment*. Within the process a resource changes states: *ResourceReceived*, *ResourceVerified*, and *ResourceTransferred*. The process *DeliverStockedResource* includes nine transactions: *InventoryReservation*, *OrderConsolidation*, *LoadsBuilding*, *ShipmentsRouting*, *CarrierSelection*, *Receiving*, *Pick*, *Pack*, and *Payment*. Within this process a resource changes states: *ResourceReceived*, *ResourcePicked*, and *ResourcePacked*.

All the patterns of the Framework are reflected to the Application level. As an illustration of Application level patterns we shall consider Transferring transaction and its relations with other objects. Transferring transaction is involved into “whole-part” relationships with its production fact *PF-Transferring* and *SourceStockedResource* process (Fig. 11). Resource states as well as transactions form a sequence by means of *beforeAfter* patterns (Fig. 12). A sequence of transactions linked by preconditions (Fig. 13) forms the whole process (*SourceStockedResource* or *DeliverStockedResource*). Some transactions change resource states as a result; therefore, proposed metamodel provides an opportunity to trace resource lifecycle within a certain process. In addition, each production fact has responsible actors (*responsibleForProductionFact* pattern) (Fig. 14).

The power of the metamodel is also provided by cardinalities and additional logic rules of objects’ associations. These parts of proposed metamodel were implemented in OWL and in the form of coded rules. Moreover, first order logic

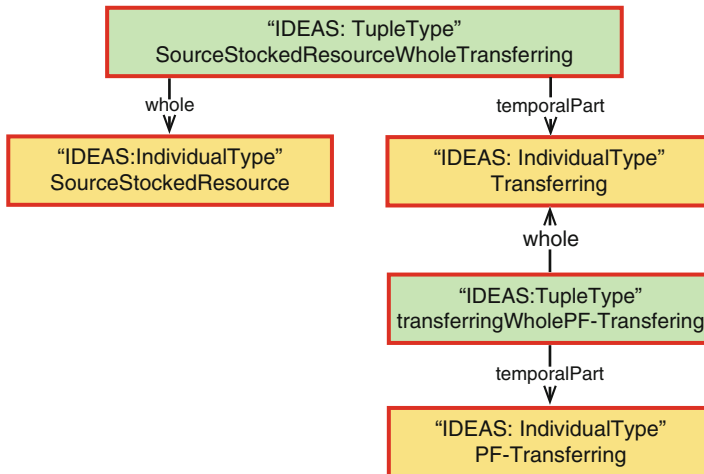


Fig. 11 wholePart pattern for Transferring transaction

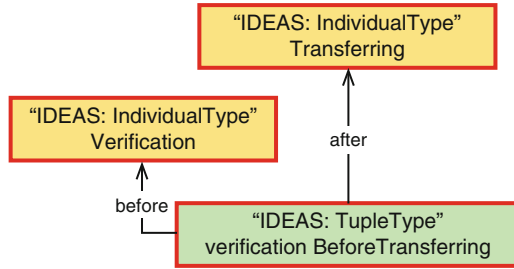


Fig. 12 *beforeAfter* pattern for *Verification* and *Transferring* transactions

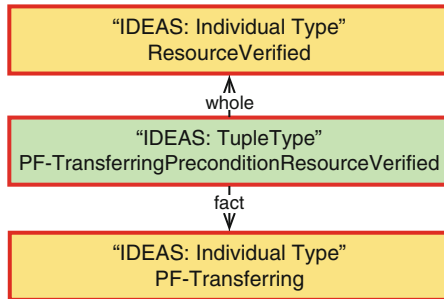


Fig. 13 *precondition* pattern for production fact of *Transferring* transaction

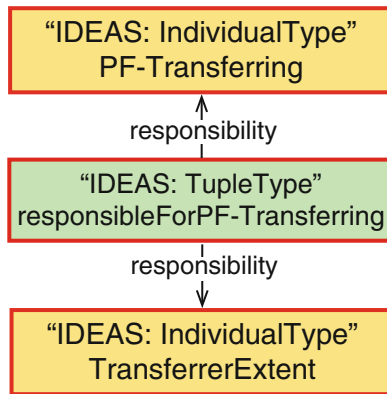


Fig. 14 *responsibleForProductionFact* pattern for production fact of *Transferring* transaction

rules were used for extracting new knowledge that is not explicitly presented in the metamodel. These rules add additional semantic power and allow building more complex queries to related data model (Sect. 5).

The Application level of created conceptual model contains 49 classes of individual objects and 71 associations (classes of relations). All created classes are the subclasses of the formal enterprise ontology [10].

At the stage of implementation (in accordance with SABiO's Development Process [13]) the ontology was presented in OWL format (dialect FULL). The OWL representation takes into account all the classes and classes of classes of objects and relations, cardinalities of object dependencies, restrictions on classes' properties, comments with the definitions of concepts, and the hierarchy of metamodel levels.

## 4 Case Study of Supply Network

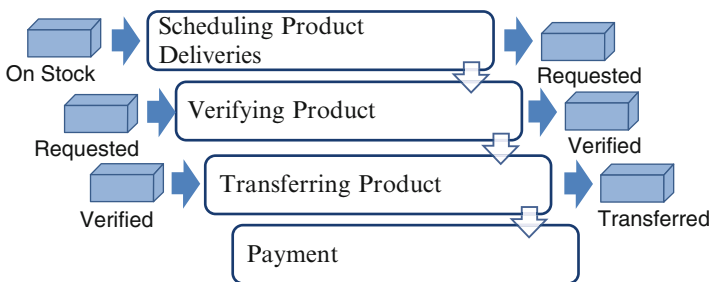
To verify completeness and coherence of the developed metamodel we apply it to the use-case<sup>8</sup> that describes a possible business scenario close to supply processes of the SCOR model.

The Fig. 15 shows the sequence of subprocesses of the whole process of supply. The case includes four stages: Scheduling Product Deliveries, Verifying Product, Transferring Product, and Payment. The first three stages change product states: On Stock, Requested, Verified, and Transferred.

Apart from process stages and product states, the use-case gives information about involved agents and their roles (Table 1).

The stages of the business process being described resemble subprocesses of S1 process of the SCOR model except Receiving Product subprocess. The flexibility of the metamodel allows avoiding fixed sequence of process stages and, therefore, applying the model to various cases.

All objects above and relevant relationships (*beforeAfter*, *wholePart*, *precondition*, *responsibleFor*, and other patterns) among them were added to the data model and presented in OWL format to make the model appropriate for further computer processing. Ultimately, OWL representation of the Operational level contains 30 instances of individual objects and 37 instances of relationships.



**Fig. 15** The structure of the business process given in the use-case

<sup>8</sup>Taken from GS1 Logistics Interoperability Model (GS1 LIM), Version 1, Issue 1.0 (2007).

**Table 1** Agents and their roles within the process of replenishment

Role	Agent
# Manufacturer	# Laithwaite's_wine
# Material supplier	# Le_Chai_au_Quai
# Manufacturer warehouse	# Laithwaite's_wine_Warehouse
# Material supplier warehouse	# Le_Chai_au_Quai_Warehouse
# Transport service provider	# HAROPA

## 5 Semantic Analysis with Proposed Logistics Ontology

The use-case gave us instances of created classes on the Operational level of the data model. According to the systematic approach for building ontologies (SABIO's Development Process [13]), we made a set of queries to the data model based on some competency questions. Competency questions are the most important questions from concerned parties' perspective. The amount and the complexity of queries the metamodel is able to answer shows its completeness and coherence, and its semantic power. Therefore, it is crucial to test the metamodel by means of competency questions.

The logic of queries building is the following. Each information pattern of the metamodel is an atomic part of more complex information item. In such a way, complex queries associated with complex information items comprise a set of atomic queries associated with basic information patterns (ref. to Sect. 3). This logic was implemented into software prototype designed on Apache Jena<sup>9</sup> platform by means of coded rules for extracting new knowledge. Table 2 contains some spread questions in the natural language. Questions 1–6 are more general while the rest questions are based on the basic Application level patterns. Evidently, the Table 2 can be extended because the set of possible queries is defined by the set of numerous combinations of information patterns. Notably, the cardinalities of associations and implemented logic rules of proposed metamodel form a “semantic glue” of complex information items.

The semantic power makes the metamodel a core for different metamodels integration. As a particular case we consider combined data metamodel of three projects of the EU 7th Framework program (FP7): EURIDICE,<sup>10</sup> iCargo, and<sup>11</sup> e-Freight.<sup>12</sup> These projects put joint efforts on development of the new generation of information systems in logistics, including the multilayered semantic metamodel [14].

The top level of the FP7 metamodel includes concepts: “Activity,” “Event,” “Role,” “Actor,” “StaticResource,” and “MoveableResource”. “Activity” denotes

<sup>9</sup>Jena Ontology API: <http://jena.apache.org/documentation/ontology/>.

<sup>10</sup>EURIDICE project: <http://www.euridice-project.eu>.

<sup>11</sup>i-Cargo project: <http://i-cargo.eu>.

<sup>12</sup>E-Freight project: <http://www.efreightproject.eu>.

**Table 2** Implemented competency questions and their associated basic information patterns

No	What is/are	Information pattern
1	The role of an agent in a process?	<ul style="list-style-type: none"> <li>• <i>processWholeActiveParticipationExtent</i></li> <li>• <i>agentParticipation</i></li> </ul>
2	The resources/products involved into a process?	<ul style="list-style-type: none"> <li>• <i>processWholePassiveParticipationExtent</i></li> <li>• <i>passiveParticipation</i></li> </ul>
3	The sequence of process transactions?	<ul style="list-style-type: none"> <li>• <i>processWholeTransactions</i></li> <li>• set of <i>processStateBeforeAfter</i></li> </ul>
4	The result of a process?	<ul style="list-style-type: none"> <li>• <i>processWholeTransaction</i></li> <li>• set of <i>processStateBeforeAfter</i></li> <li>• <i>transactionWholeProductionFact</i></li> <li>• <i>factPreconditionProductState</i></li> </ul>
5	The detailed lifecycle of a product?	<ul style="list-style-type: none"> <li>• set of <i>resourceStateBeforeAfter</i></li> <li>• <i>productStatebeforeAfter</i></li> </ul>
6	The responsible agent for a transaction?	<ul style="list-style-type: none"> <li>• <i>responsibleForProductionFact</i></li> <li>• <i>agentParticipation</i></li> </ul>
7	The role of an agent in Source Stocked Resource process?	<ul style="list-style-type: none"> <li>• <i>agentParticipation</i></li> <li>• <i>sourceStockedResourceWhole-ActiveParticipationExtent</i></li> </ul>
8	The resources/products involved into Source Stocked Resource process?	<ul style="list-style-type: none"> <li>• <i>sourceStockedResourceWhole-PassiveParticipationExtent</i></li> <li>• <i>passiveParticipation</i></li> </ul>
9	The previous transaction for Transferring transaction?	<ul style="list-style-type: none"> <li>• <i>verificationBeforeTransferring</i></li> </ul>
10	The result of Transferring transaction?	<ul style="list-style-type: none"> <li>• <i>transferringWholePF-Transferring</i></li> <li>• <i>PF-TransferringPrecondition-ResourceTransferred</i></li> </ul>
11	The previous state for Resource Transferred state?	<ul style="list-style-type: none"> <li>• <i>resourceVerifiedBeforeTransferred</i></li> </ul>
12	The responsible agent for Verification transaction?	<ul style="list-style-type: none"> <li>• set of <i>verificationWholePF-Verification</i></li> <li>• <i>responsibleForPF-Verification</i></li> </ul>

an action and is connected with “Actor” via “hasProvider” and “hasConsumer” relations, “Event” means something that happens and has no time duration. “Actor” performing an “Activity” has “Role” that is expressed by means of “hasRole” relationship. And finally, “Activity” is associated with resources (“StaticResource” and “MoveableResource”) by “hasStaticResource” and “hasMoveableResource” relations.

Following the meaning of enumerated concepts we integrated FP7 metamodel with our metamodel. “Activity” can be considered as a subclass of *Transaction* class, “Actor”—a subclass of *Agent*, and “Role”—a subclass of *ActiveParticipationExtent* (Fig. 16). “Event” has the meaning of *CoordinationFact*, “StaticResource” and “MoveableResource” are subclasses of *Resource* (Fig. 17). Relationships of FP7 metamodel can also be integrated as following. “hasProvider”

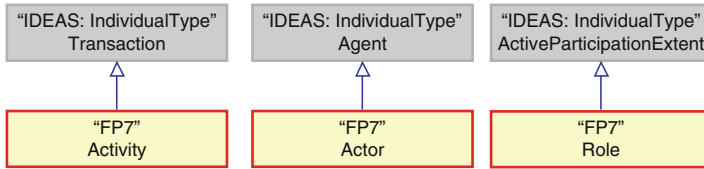


Fig. 16 “Activity,” “Actor,” and “Role” integrated with the metamodel

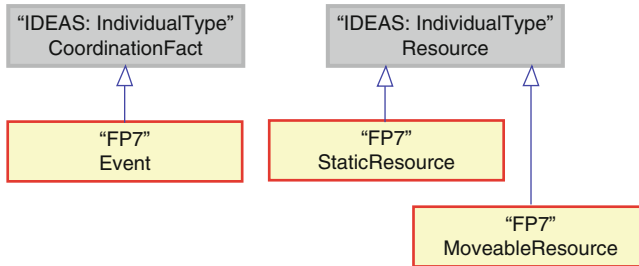


Fig. 17 “Event,” “StaticResource,” and “MoveableResource” integrated with the metamodel

and “hasConsumer” become subclasses of *responsibleForProductionFact* and *responsibleForCoordinationFact* classes, correspondingly, “hasRole” a subclass of *agentParticipation*, and “hasStaticResource” and “hasMoveableResource” are subclasses of *passiveParticipation* class. Thus, all top level classes and the main associations of the combined FP7 metamodel were easily integrated with the proposed metamodel. Together with other subclasses of the FP7 metamodel, obtained result of metamodels integration can be considered as a particular metamodel of the open supply network supported ontology-based data exchange.

Thus, the strong semantics of proposed logistics ontology facilitates integration of metamodels. Moreover, it helps to reveal semantic gaps in the integrated conceptual models. Following the constraints of proposed metamodel, we found that complex information patterns cannot be formed on the basis of initial FP7’s data metamodel. For example, the information patterns answering questions: “What are results of an activity?,” “How resources/products were transformed within a process/activity?,” “What is the sequence of activities”, and so on cannot be built. Moreover, some redundant associations were revealed in the initial metamodel during the integration process.

In such a way, the proposed metamodel can be a platform for further integration of different logistic standards and existing ontologies due to its semantic power. Thus, it forms the platform for semantic interoperability.

## 6 Conclusions

The article proposes the logistics ontology and its implementation—the data metamodel for the domain of logistics built upon BORO foundational ontology and the formal enterprise ontology [10]. In the research we extracted consistent conceptual model from the SCOR standard using the BORO methodology. On the other hand, we used the formal enterprise ontology to build semantically reach data metamodel describing the lifecycle of products and providing the transparency of supply networks. Proposed ontology-based metamodel is aimed at serving a basis for the integration of different ontologized logistic standards and particular data metamodels that are built upon the SCOR standard. The top level of the metamodel contains 38 classes of individual objects and 48 types of associations, the logistics-specific level—49 classes of individual objects and 71 types of associations. For the validation and testing of proposed metamodel we used simulated data, to wit: 30 instances of individual objects and 37 particular associations were specified.

At the stage of validation the metamodel was applied to ordinary use-case close to supply processes of the SCOR model. The metamodel was tested by means of the set of queries in order to prove its completeness and coherence and to evaluate its semantic power. For extracting new knowledge from the ontology-based data model the software prototype was implemented on Apache Jena platform. To show the suitability of proposed logistics ontology for the resolution of interoperability problem, the developed metamodel was integrated with another existing metamodel of the logistics domain. Developed metamodel helped to reveal semantic gaps in the integrated metamodel as well as allowed to perform the integration without any changes in the integrated metamodel.

In the nearest future both created logistics ontology and the data metamodel as its implementation will be reinforced by the shared core of leadership standards in logistics (GS1 and UN/CEFACT).

**Acknowledgements** This research is partially supported by LATNA Laboratory, NRU HSE, RF government grant 11.G34.31.0057, and by the CLASSE (“Les Corridors Logistiques: Application a la Vallee de la Seine et son Environnement”) project of the Grand Research Network of in Upper Normandy (Grand Réseaux de Recherche de Haute-Normandie).

## References

1. European Interoperability Framework for PAN-European eGovernment Services. European Communities (2004). ISBN 92-894-8389-X
2. Partridge, C.: *Business Objects: Re-Engineering For Re-Use*. Springer, Berlin (2000)
3. Partridge, C.: The role of ontology in semantic integration. In: *Second International Workshop on Semantics of Enterprise Integration at OOPSLA 2002*, Seattle (2002)
4. Bjeladinovic, S., Marjanovic, Z.: A comparison and integration of ontologies suitable for interoperability extension of SCOR model. In: Bogdanova, A.M., Gjorgjevikj, D. (eds.) *ICT Innovations 2014. Advanced in Intelligent Systems and Computing*, vol. 311, pp. 75–84. Springer, Cham (2015)



5. Zdravkovic, M., Panetto, H., Trajanovic, M., Aubry, A.: An approach for formalizing the supply chain operations. *J. Enterp. Inf. Syst.* **5**, 401–421 (2011)
6. Magdalenic, I., Vrdoljak, B., Schatten, M.: Mapping of core components based e-business standards into ontology. In: Sanchez-Alonso, S., Athanasiadis, I.N. (eds.) *Metadata and Semantic Research. Communications in Computer and Information Science*, vol. 108, pp. 95–106. Springer, Berlin (2010)
7. Al Asswad, M.M., Al-Debei, M.M., de Cesare, S., Lycett, M.: Conceptual modeling and the quality of ontologies: a comparison between an object-role modeling and the object paradigm. In: *Proceedings of the 18th European Conference on Information Systems*, pp. 1–12. Association for Information Systems (AIS) (2010)
8. International Defense Enterprise Architecture Specification Group: <http://ideasgroup.org/>
9. Supply Chain Operations Reference (SCOR) Model Version 9.0. Supply Chain Council (2008)
10. Poletaeva, T., Babkin, E., Abdulrab, H.: Ontological framework aimed to facilitate business transformations. In: Guizzardi, G., Pastor, O., Wand, Y., de Cesare, S., Gailly, F., Lycett, M., Partridge, C. (eds.) *Proceedings of 1st Joint Workshop Onto.Com/ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering, Rio de Janeiro, 22–25 Sept 2014*, vol. 1301. CEUR-WS (2014)
11. Bailey, I., Partridge, C.: Working with extensional ontology for defence applications. In: *Ontology in Intelligence Conference Proceedings*, Fairfax, VA (2009)
12. Dietz, J.: *Enterprise Ontology ? Theory and Methodology*. Springer, Berlin/Heidelberg (2006)
13. Falbo, R. de A.: SABiO: systematic approach for building ontologies. In: Guizzardi, G., Pastor, O., Wand, Y., de Cesare, S., Gailly, F., Lycett, M., Partridge, C. (eds.) *Proceedings of 1st Joint Workshop Onto.Com/ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering, Rio de Janeiro, 22–25 Sept 2014*, vol. 1301. CEUR-WS (2014)
14. EURIDICE Integrated Project ICT 2007-216271. Workpackage 12: Domain Knowledge Formalization, Version 2.0 (2009)