**Journal of Supply Chain Management Science**

https://journals.open.tudelft.nl/jscms

http://dx.doi.org/10.18757/jscms.2020.5444

# Seamless interoperability in logistics by ontology alignment

Majid Mohammadi [a,b], Wout Hofman [c], Yao-Hua Tan [a]

[a] *Faculty of Technology, Policy and Management, Delft University of Technology, 2628 BX Delft, The Netherlands*
[b] *The Jheronimus Academy of Data Science, Eindhoven University of Technology, 5211 DA 's-Hertogenbosch, The Netherlands*
[c] *The Netherlands Organization for Applied Scientific Research (TNO), 2600 AA Delft, The Netherlands*

Article history: received 07-12-2020, accepted 28-12-2020

**Abstract** – This paper is dedicated to applying ontology alignment systems to the heterogeneity problem in logistics. The primary motivation for doing so is to enable interoperability among different IT systems in logistics, all with their own database scheme. We first analyze different standards for logistics interoperability, which are implemented by XML schema definition (XSD) or Electronic Data Interchange (EDI) with implicit and inconsistent semantics across these open standards. To analyze the applicability of ontology alignment, we create two ontologies from two well-known standards that are rich in terms of semantic relations between entities. These ontologies are subjected to the state-of-the-art alignment systems, applying matching with and without logistics background knowledge. Our experimental analyses show that the alignment using background knowledge with some annotations finds better mappings between the given logistics standards and is thus applicable to real-world situations under particular conditions.

**Keywords:** Interoperability; logistics; ontology alignment

## 1. Introduction

The supply and logistics sector consists of many large and Small and Medium-sized Enterprises (SMEs) with limited IT skills and knowledge. Its total turnover in the EU only is estimated at 878 billion Euro in 2012[1] with over 1.2 million enterprises (Satta et al. 2011). Besides proprietary developed software, these enterprises can choose to use Commercial-Off-The-Shelve (COTS) software from over 200 different suppliers[2], each of which has its proprietary database scheme that can even be customized by the users. The challenge is and has been to integrate business processes of enterprises and their supporting IT solutions to reduce costs, increase efficiency, and enable effective ways of operation. The development of open standards addresses this challenge. Although these standards were developed, their implementation did not solve heterogeneity among different standards or logistics IT systems. Different implementation guides of (different versions of) open standards have been developed (Hofman 2019), leading to implementations that are only interoperable with additional efforts and costs. These implementation guides support process interoperability (Wang et al. 2009), which implies they will support a particular function for interconnecting business processes. There are also different open standards providing the same business functionality, e.g., a transport order developed by UN/CEFACT or one of the Uniform Business Language (UBL). Commercial organizations provide transformation services between various data standards to address differences in implementation guides of different open standards with identical or similar functionality.

Applying open standards and commercial transformation services reduces the transformation challenge, but the development of implementation guides of these open standards still takes too much time for developing supply chain innovations like agility, resilience (Wieland and Marcus Wallenburg 2013), and synchromodal planning (Giusti et al. 2019). In other words, the use of different standards or implementing the same standards differently introduces another heterogeneity challenge, though not more severe than having no standards, that still needs to

[1] https://ec.europa.eu/transport/themes/logistics-and-multimodal-transport/logistics_en

[2] https://www.capterra.com/logistics-software/

be addressed properly. To reduce these development and implementation time for interoperability between any two organizations, this paper explores the application of ontology alignment.

The holy grail of ontology alignment in general is to create semi-autonomous alignments between database schemes of different organizations, thus enabling what one could call ‘plug and play’[3]: Plug a database scheme into an open data sharing infrastructure and be able to share data with relevant business partners. Plug and play requires an open data sharing infrastructure providing standardized services (Hofman and Dalmolen 2019).

There are several issues in applying ontology alignment to enable interoperability between different logistics stakeholders. First, the number of enterprises is significant, making the pairwise alignment between every two parties very time-consuming. For instance, if the aim is to enable interoperability among 1.2 million logistics parties in the EU only, then we need to execute an alignment system $1.44 \times 10^{12}$ times. This number increases even more, if several messages have to be implemented pairwise, e.g., a transport order, booking, invoice, and tracking status message. Another challenge is that standards are usually modeled by XML schema definition (XSD) that conveys no or limited semantics of entities in the associated standards. As a result, it is required to create at least two ontologies from these XSD models in order to be able to experiment the applicability of ontology alignment. In addition, the standards have different level of granularities: some are mode specific, e.g., air or rail, and some are more general and encompass different modes from a higher view.

This paper studies the applicability of ontology alignment to address interoperability between heterogeneous IT systems in supply and logistics. We first review the standards used in logistics that aimed at solving the interoperability problem. We then investigate two specific standards and create ontologies based on their XSD models. We finally use ontology alignment to find the shared entities of ontologies and experiment the usefulness of such an approach to address the interoperability in logistics.

The remainder of this paper is organized as follows. Section 2 contains the existing ways of dealing with heterogeneity in logistics. Section 3 is dedicated the creation of two ontologies from two standards in logistics, while Section 4 is dedicated to ontology alignment concepts and the way ontologies should be matched in logistics. The experiments regarding the applicability of ontology alignment in logistics are presented in Section 5, and we conclude the paper in Section 6.

## 2. Interoperability in Logistics

Logistics interoperability has to be considered in the context of collaborating business processes of stakeholders, applicable standards to support interoperability between these business processes, and the technical paradigm for data sharing applied. Any choices made in these different areas may affect the applicability of ontology alignment in logistics. Therefore, we will briefly present an overview of these aspects.

### 2.1. Logistics Business Processes

International trade and logistics are characterized by moving cargo from one location to another with a particular transport means, (temporary) storage of this cargo, and authorities governing these cargo flows from different legal perspectives like safety, security, and VAT compliance. Each modality and each cargo type have their specific characteristics. For instance, bulk cargo considers weights and volumes and containerized cargo a container with its size and type and container identification. Furthermore, sea containers have other characteristics than containers used for air transport. The latter are called Uniform Load Devices and exactly fit into an airplane. Different transport modalities also use different infrastructures with their hubs, have different transport documents, and so forth. There are also different enterprises and authorities involved, like a food and drug inspection agency for transport of agricultural cargo, coastal police for vessel movements, and air traffic control for air transport.

Modalities may have standardized the structure of data sets they share for digitization of their business processes (see next section). Each individual organization collaborating in logistics chains will have its own IT system with its own internal data structure. Interoperability is about integrating these heterogeneous IT systems. It implies that each organization will have its implementation of for instance a transport order.

[3] The Digital Transport and Logistics Forum (DTLF)

These internal data structures have to be matched with the ones for modalities or any de facto structures used by their customers or major service providers. Since the number of logistics enterprises is large, business process integration of all collaborating organizations is a challenge.

Two collaborating organizations will share multiple data sets for business process digitization, like a booking, a transport order – and plan, and an event for reporting the progress. Thus, integration complexity increases. It can also increase in solutions for all variants of logistics chains are developed. Furthermore, integration complexity increases by the number of logistics enterprises, which runs into million globally, all (inter)national legislation with their governing authorities, and changes in trade agreements that have impact on data requirements of authorities.

Ontology alignment might provide a solution to reduce integration complexity. Another approach for complexity reduction that we will investigate together with ontology alignment, is to abstract from these chains and specify bilateral data sharing (Schonberger et al. 2010). Each chain is constructed by its links of any two collaborating organizations and the outsourcing rules applied by each individual organization. This reduces complexity and still enables (dynamic) chain composition. Any interactions between these collaborating organizations can be modelled as a business process choreography (GROUP et al. 2011), where a semantic model specifies all data that can be shared. The semantic model, which can be represented by an ontology, is called a Canonical Information Model (Erl 2005). The choreography supports business functionality as developed by the DTLF:

- Publish, search, and find logistics capacity. By posting a particular goal, a customer can find available transport -, storage – or other type of logistics capacity. The capacity might be offered by timetables of for instance vessels (called: voyage) or trains.
- Booking and ordering. Whenever capacity or a logistics service provider has been found, booking or a request for quotation can be made. This can result in a framework contract followed by orders for individual shipments or a booking can directly be confirmed as an order.
- Supply chain visibility. This is about sharing relevant milestones of the progress of a logistics service, e.g., loading, departure and (estimated) arrival of a truck at its destination.

Each functionality is supported by interactions for data sharing, where these interactions are of a type. For in- stance, booking and ordering is supported by a booking, a booking confirmation, a transport order, and a plan. These interaction types specify the minimal data that needs to be shared and maximal that can be shared in the context of the choreography. For instance, a transport order should at least contain one cargo item and two locations (acceptance and delivery) with their respective time windows for acceptance and delivery. This minimal – and maximal data set is formulated in terms of the semantic model. A first version of a semantic model has been constructed in various EU funded projects. This will be elaborated at a later stage in this paper.

## 2.2. Open Standards and Their Implementation

There are different logistics standards to enable the interoperability in the domain. There are different classifications for standards that categorize them into different levels. For instance, the European interoperability framework (EIF) (Union 2017) has four levels: technical, semantical, organizational, and legal. A more thorough model is the levels of conceptual interoperability model (LCIM) (Wang et al. 2009) that identifies six different levels, to which the levels of EIF can be mapped. Figure 1 plots the standards and their pragmatic application in logistics to the LCIM. The figure shows that semantic models generate syntactical standards and can be used to create implementation guides (IGs). The technical interoperability is based on communication protocols so that the heterogeneity is trivial. To the best of our knowledge, there is no interoperability model for the top two levels, dynamic and conceptual interoperability. Therefore, three remaining LCIM levels are only explained in the following:

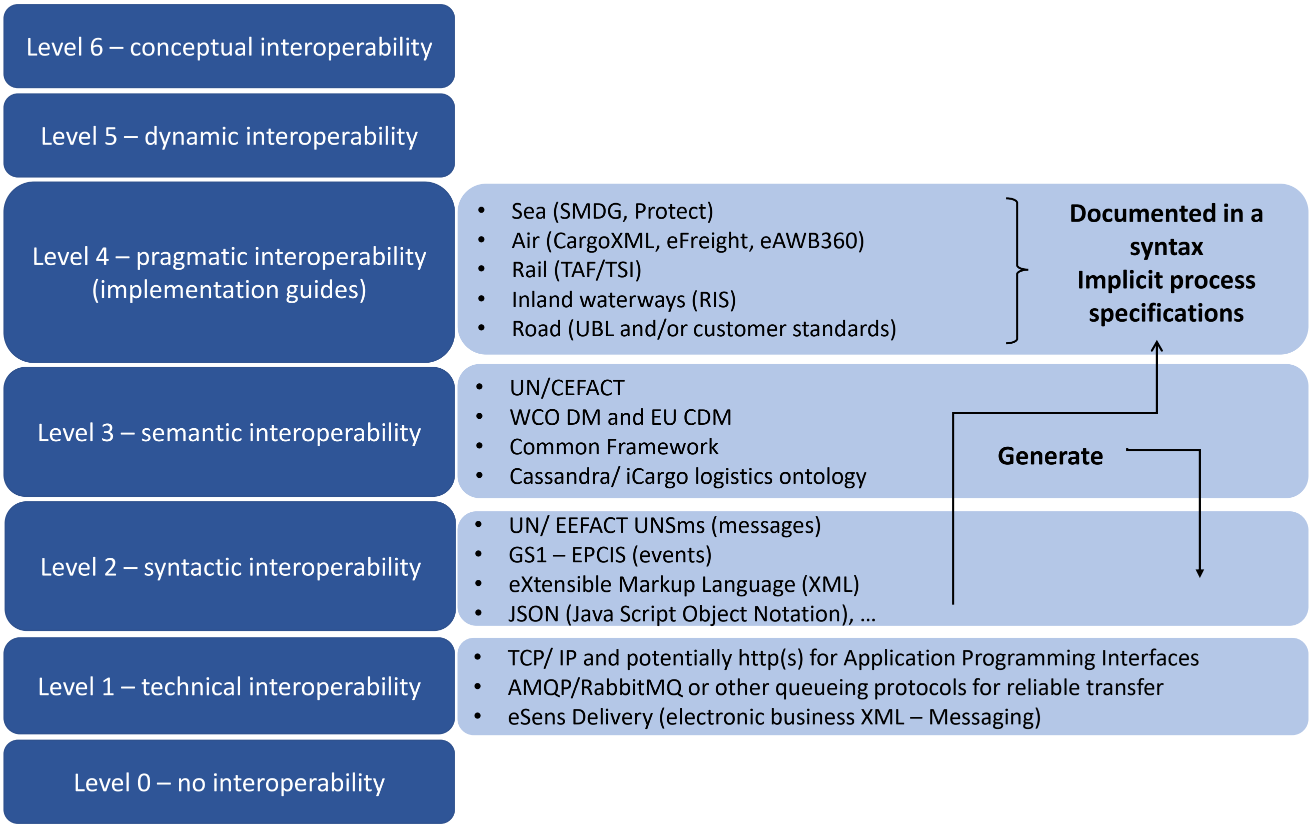


Figure 1. Overview of logistics standard (Hofman 2019).

- **Syntactic Interoperability**: This class refers to the structure of data during exchange. The syntax of a message governs the use of protocols in technical interoperability level. XML and JSON are shown as potential syntaxes for syntactical interoperability. Another option is UN/CEFACT United Nations Standard Messages (UNSMs) as technical data structures representing specific interaction types, e.g., there is a UNSM for a transport order. These UNSMs use the EDIfact (Electronic Data Interchange for administration, commerce, and trade) as syntax (ISO 1988). UNSMs have lots of configuration options; they are generic.
- **Semantic Interoperability:** This class refers to the semantic representation of the data modelled, for instance, by unified modelling language (UML) or ontology web language (OWL). The models are mostly used to develop message structures for sharing data at syntactic level, for instance to automatically generate UNSMs or XSDs (XML Schema Definitions).
- **Pragmatic Interoperability:** it means that two or more stakeholders integrate their business processes. They start from existing processes and most often replace current procedures, which are mainly paper-based, with electronic versions. Business process integration can be based on level 2 standards; most often these collabo- rating business partners do not have access to data models (if they are available). In case the use existing open standards like UNSMs or XSDs, they configure these to their requirements resulting in so-called implementation guides of these open standards. In case these implementation guides are constructed by more than two organizations, they can be called community implementation guides, for example the ones specified for rail or sea. These community implementation guides can also be a basis for any two organizations constructing their bilateral implementation guides.

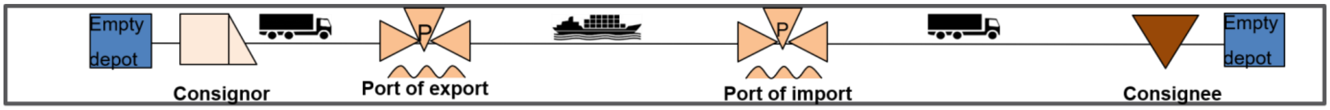


Figure 2. An example of the physical activities for a shipment from a consignor to a consignee.

Thus, analysis of (open) standards and their implementation learns that 'closed' data sharing solutions are con- structed. Open standards do not necessarily lead to open ways of data sharing, meaning that it is easy to on-board organizations. On the one hand, implementation guides will be constructed and on the other hand they have to be implemented by matching to heterogeneous IT systems of organizations. Thus, having open standards only does not necessarily reduce enterprise interoperability complexity in logistics.

## 3. Semantic Logistics Models

To In this section, two ontologies based on two logistics data models, electronic CMR[4] and shipping instruction (SI), are created that also include the semantics among their different entities. eCMR is a standard that is used in the road transportation, while SI is for a transportation by sea. To show the importance of matching these two data structures, Figure 2 shows a typical transportation trip of a cargo from an origin to a destination. First, the consignor ships a cargo to a port, named here as the port of export. In this example, the transportation to the port of export is by road so that their associated shipping information is modeled by eCMR. Then, the cargo is transshipped to a port of import by sea, whose transshipping information is modeled by SI. Then, from the port of import, the cargo will be sent to the consignee by means of road, that again necessitates to store the information by the eCMR model. In this simple example, we need to transform the data first from eCMR to SI and then from SI back to eCMR. As a result, the alignment of these models is very essential in logistics. In the following, we first create two ontologies and then explore the alignment of the two ontologies together.

### 3.1. Electronic CMR Ontology

The CMR is a United Nations convention that concerns with various legal issues concerning the transportation of a cargo by road. As of 2017, CMR has been ratified by most of the European states. The International Road Transport Union (IRU) developed a standard CMR waybill according to the CMR. As of 2008, it is also feasible to use an updated electronic consignment note, called eCMR. We create an ontology based on a subset of eCMR by using the terminology applied to the eCMR standard of UN-CEFACT, which is more than a data carrier and is able to contain semantic relations between entities.

The classes of the eCMR as well as their related properties in the created ontology are as follows (class names are identified by italic and bold characters and property names only by italic characters):

- ***LogisticsLocation*** includes the locations in a logistics trip and contains basic information for a location, such as name and country. The following two classes inherit from this class:
  - ***CarrierAcceptanceLogisticsLocation*** is the location where a cargo is picked up by a carrier.
  - ***ConsigneeReceiptLogisticsLocation*** is where a cargo will be delivered to a consignee.

- ***TradeParty*** represents different parties in a transaction. It has three main subclasses associated to different parties:
  - **Consignor** is the one that orders a consignment and must determine locations of acceptance and delivery.
  - **Consignee** is the one to whom a cargo must be delivered.
  - **Carrier** is the party that conducts the shipment by picking up the cargo from an acceptance place and delivering to a consignee receipt location.

[4] It stands for Convention on the Contract for the International Carriage of Goods by Road

- ***SupplyChainConsignmentItem*** includes the items in the shipment. This class has several relationships with other classes as:
  - The items are inside a transport cargo. Hence, we use object property *isInsideCargo* that relates this class to ***TransportCargo***.
  - The items are placed in a logistics package. Thus, we use object property *isPackagedIn* to relate this class to ***LogisticsPackage***. In addition, the packages must have shipping marks that makes ***LogisticsPackage*** have another object property *isMarkedAs* to link it to ***LogisticsShippingMarks***.
- ***SupplyChainConsignment*** is the main class in eCMR that represent the shipment process. It has multiple object properties that relates it to different classes and provide the necessary information for shipping a cargo. These object properties are as follows:
  - *includes* that relates it to *SupplyChainConsignmentItem*.
  - *isDeliveredTo*, *isCarriedBy*, and *isOrderedBy* are with respect to the three different trade parties, *Consignee*, *Carrier*, and *Consignor*, respectively.
  - *isPickedUpAt* and *isDroppedOffAt* relate the class to *CarrierAcceptanceLogisticsLocation* and *ConsigneeReceiptLogisticsLocation*, respectively.
  - *hasTransportInstruction* and *hasDeliveryInstruction* relate this class to transport instructions for the shipment and delivery, respectively. These instructions are necessary according to eCMR.

Figure 3 plots the eCMR ontology visualized by VOWL (Lohmann et al. 2016). The classes are shown by circles and object properties are the labels of arrows of the two corresponding classes.

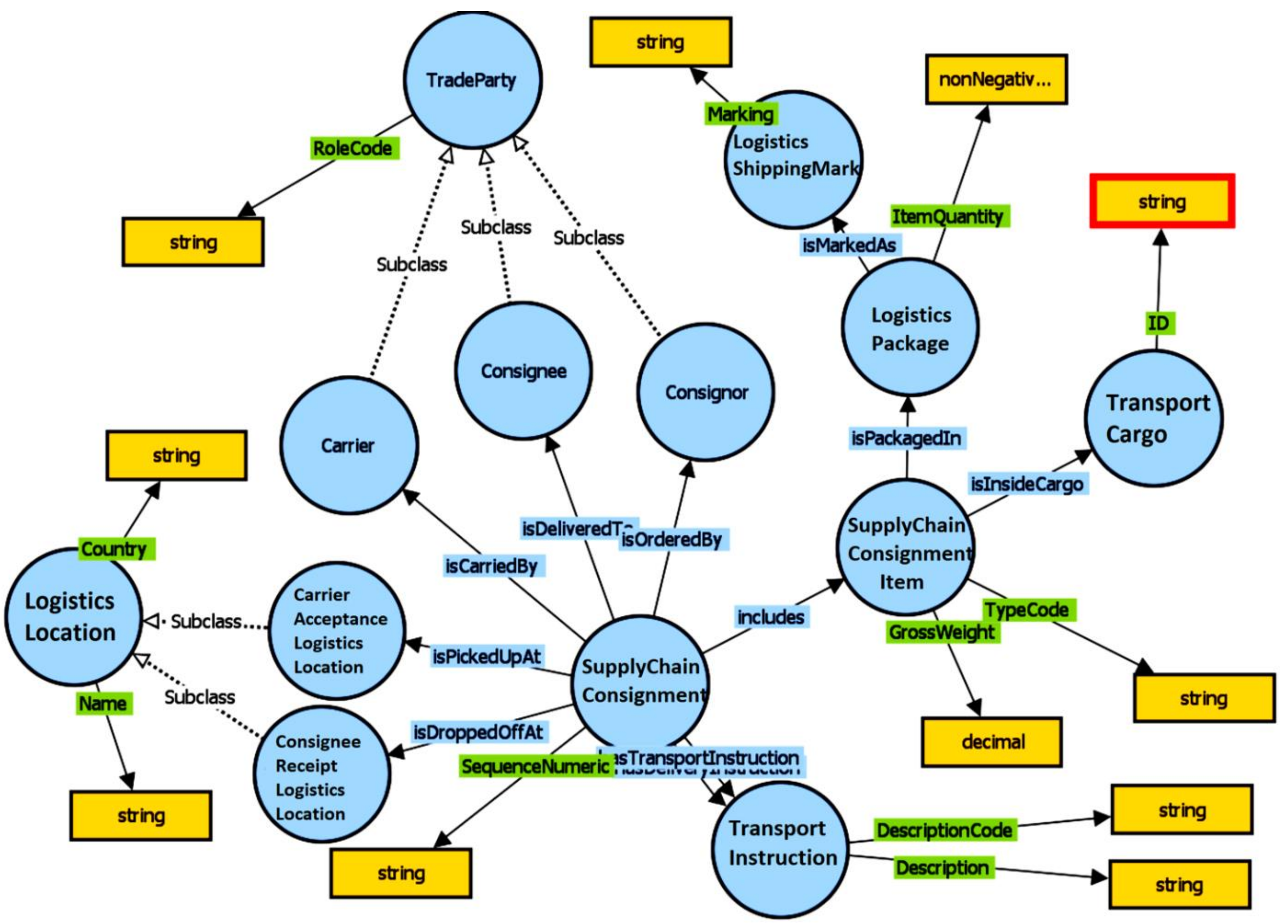


Figure 3. The eCMR ontology.

### 3.2. Shipping Instruction Ontology

A shipping instruction (SI) is a document, provided by a customer to a carrier, containing the details of a cargo to be shipped by sea and the requirements for its physical transportation. The fields in the document are the building blocks for creating the associated ontology that is based on the terminology used in SI. We particularly utilize the interface of a booking for shipment of containers by sea. The created ontology has a main class, ***ShippingInstruction***, that associates with other classes with proper object properties. These classes and properties are discussed in the following:

- It contains the information of locations where a container is picked up or delivered to. Thus, there are two object properties, *isPickedUpFrom* and *isDroppedOffAt* that associate it to ***PlaceOfReceipt*** and ***PlaceOfDelivery***. Each of these classes inherits from **place** that contains basic information of a place such as the name of the city and the United Nation location code (*CityUNLocationCode*). In addition, it contains the ports where a container is loaded and discharged, represented by classes ***PortOfLoad*** and ***PortOfDischarge*** that are related to this class by object properties *isLoadedIn* and *isDischargedAt*, respectively. These classes also inherit from another class, named ***Port***, that stores the basic information of a port.
- Each ***ShippingInstruction*** contains three different parties that are modeled as classes with names ***Forwarder***, ***Shipper***, and ***Consignee*** that is related to with *isOrganizedBy*, *isSubmittedBy*, and *isDeliveredTo*, respectively.
- It also specifies the ***container*** that must be shipped. Thus, a class with the same name is created that is related to the SI class with object property *carry*. Besides, each container itself *includes* several ***CargoLineItem***, each of which represents a particular item in the container.
- According to SI, three different notification to three parties must be sent. The notification to parties is modeled with class ***NotifyParty***, that has three subclasses, ***FirstNotifyParty***, ***SecondNotifyParty***, and ***ThirdNotifyParty***, related to ***SI*** with object properties *FirstNotify*, *SecondNotify*, and *ThirdNotify*.

Figure 4 plots the SI ontology visualized by VOWL (Lohmann et al. 2016).

## 4. Ontology Alignment: Direct and Indirect Matching

This section contains the core concepts of ontology alignment, as well as reviews the direct and indirect matching and their suitability for the logistics domain.

A matching of a concept from an ontology to one in the other is called a correspondence, and the set of all correspondences between two given ontologies are called an alignment. The following definitions present these two important concepts in ontology alignment.

**Definition 4.1 (Correspondence (Euzenat et al. 2007)).** A correspondence between two ontologies $O$ and $O'$ is de- fined as a set of 4-tuples:

$$< e, e', r, d >,$$

where

$e$ is a concept or entity, e.g., class, property, or instance, from the first ontology;

$e'$ is an entity form the second ontology;

$r$ is the type of relation between two entities, e.g., equivalence, subsumption;

$d \in [0, 1]$ is the confidence of the matching.

**Definition 4.2 (Alignment (Euzenat et al. 2007)).** An alignment is the typical outcome of an ontology matching system and consists of several correspondences between different entities of two given ontologies.

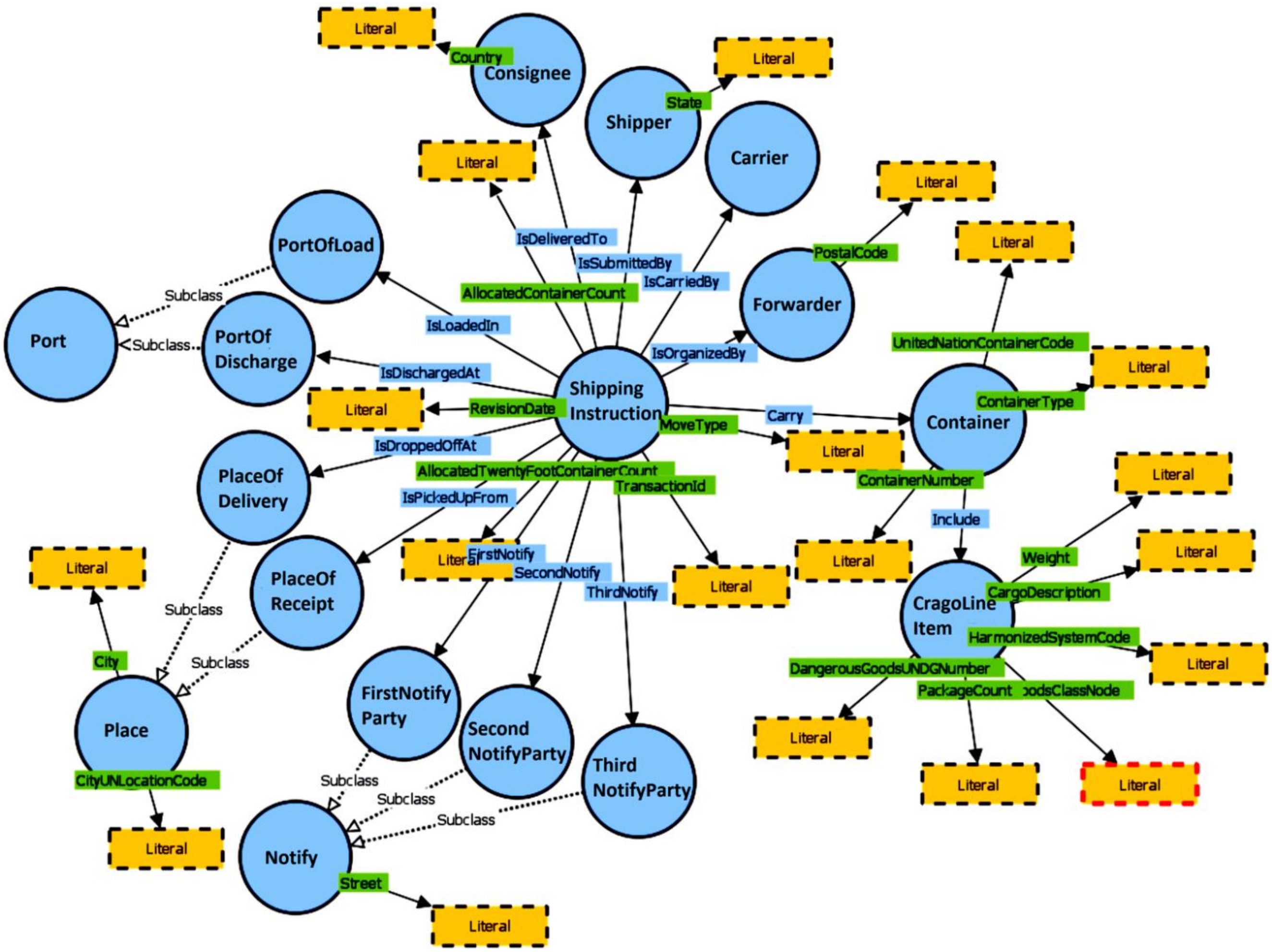


Figure 4. The shipping instruction ontology.

Generally speaking, an ontology alignment system is a software that takes two ontologies as the inputs and uses matching techniques to generate the alignment between ontologies in question. Besides, it often uses some resources such as a dictionary and requires some parameters for generating final correspondences. Figure 5 visualizes the general inputs and output to a matching system in the alignment process. In this figure, $O$ and $O'$ are two ontologies that are matched by a matching system and $A$ is the alignment generated by the system.

In some domains like biomedical, there are some ontologies that contain general terms in the domain. These ontologies that are called upper ontologies (also known as a top-level ontology, upper model, or foundation ontology) (Mascardi et al. 2007) can help increase the quality of matching between two ontologies from a domain. One way to use such upper ontologies is to first match each of the ontologies to the upper ontology, and then finding their related correspondences based on their matching with the upper ontology. This type of alignment is called indirect matching, that is visualized in Figure 6. In this figure, the ontologies $O$ and $O'$ are first matched to an upper ontology and generate two alignments $A'$ and $A''$. Then, a composition module is used to identify the alignment between $O$ and $O'$ based on $A'$ and $A''$.

Indirect matching is much more useful for matching ontologies in logistics, since the direct alignment of ontologies cannot necessarily provide reliable outcome. The use of an upper ontology in logistics can potentially increase the quality of outcome and make ontology alignment systems be applicable to real-world logistics challenges. In addition, there are millions of logistics enterprises that require to conduct transactions with other logistics companies. If the direct matching is employed, we need to align pairwise the data model of every two logistics enterprises. However, with indirect matching it is only required to align each data model to the upper ontology and the transactions can be done on-the-fly. In the next section, experiments show that the indirect matching of ontologies can bear much more fruitful outcome.

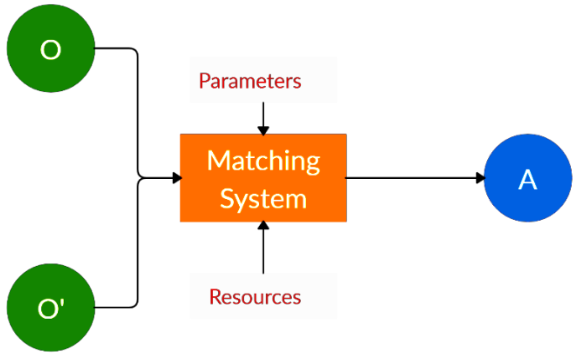


Figure 5. The direct ontology alignment process of two ontologies O and O'.

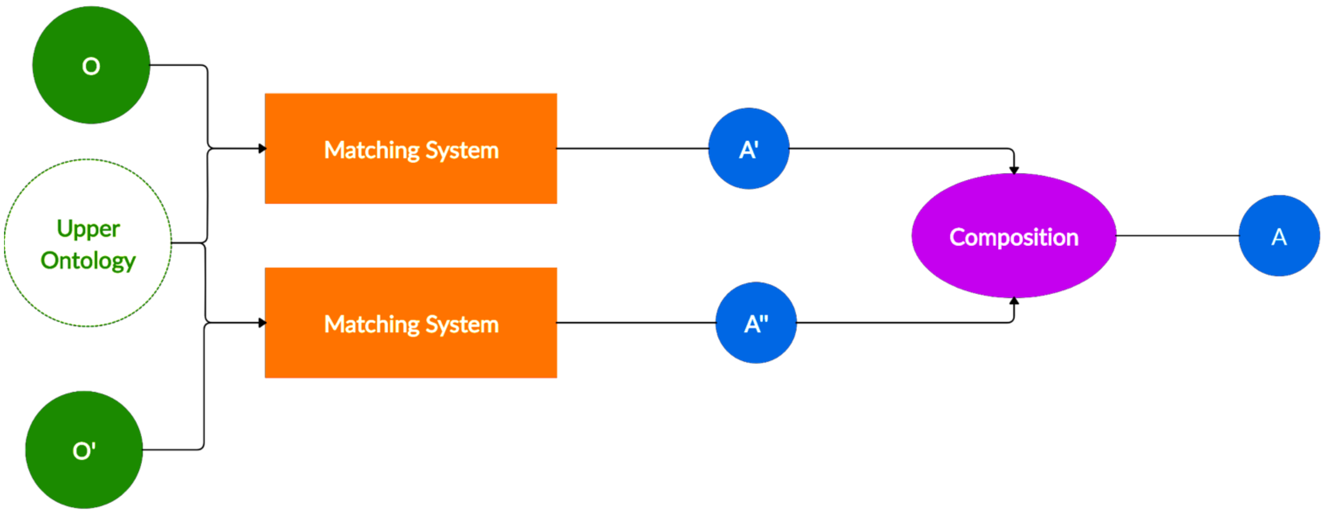


Figure 6. The indirect matching of two ontologies O and O' via an upper ontology by aid of a composition module.

## 5. Experiments

This section introduces the experiment where ontology alignment systems are applied for aligning two logistic ontologies. There are still choices to be made with respect to the experiments that will be discussed first.

### 5.1. Choices for The Experiments

Applying ontology alignment systems to logistic ontologies is different from that in the OAEI for some reasons. First of all, in spite of several efforts on using and developing ontologies in logistics (Daniele and Pires 2013, Glockner and Ludwig 2017), ontologies are not common in supply and logistics, probably due to its difficulty to develop or use them compared to other schema definition standards such as XSDs. In order to conduct experiments by using ontology alignment, open standards, their implementation guides, and database schemes have to be transformed into ontologies to enable alignment. Secondly, the following alignment choices need to be considered:

- Database scheme alignment – one could consider the alignment between database schemes of different organizations. This option is not considered feasible, since databases provide more functionality than interoperability between two organizations; they support an organization in its business.
- Functional view alignment – this option considers creating a functional view of for instance a transport order on two database schemes that will be aligned. If ontology alignment would provide optimal results, this would be an ideal situation, since it does not require any formulation of open standards. It is however also complex, while it requires to align many structures, all using potentially different terminology.

- Open standard alignment – alignment of two open standards. This could be a first start which does not require any involvement of organizations (yet). Open standards are publicly available. However, the development of an ontology from an open standard might be complex, depending on the supported functionality. An open standard for a transport order may for instance cover all transport modalities and all types of cargo.
- Implementation guide alignment – alignment of implementation guides of an open standard. For this purpose, organizations will have to provide their implementation guides.
- Alignment with a Canonical Information Model –integration of IT applications of one organization can be via an upper ontology. This approach can also be applied for external integration, i.e., between IT applications of different organizations. It requires time for constructing an upper ontology for logistics, but in case the upper ontology can be used for automatic alignment between functional views of database schemes, it will support the 'plug and play'. There are different options using an upper ontology, like:
  - Alignment of a functional view with the upper ontology;
  - Alignment of an open standard with the upper ontology;
  - Alignment of an implementation guide of an open standard with the upper ontology.

For ontology alignment, the upper ontology acts as background knowledge that can boost the alignment out- come. We can conduct the experiment by aligning implementation guides of open standards with an ontology that has been developed in EU funded projects. This experiment can be completely controlled. The ontology that acts as an upper ontology is called LogiCO[5], and an implementation guide of an existing open standard will be produced that is expected to contain concepts represented by LogiCO. As a result, for the experiment, we use the notion of indirect matching discussed in the previous section.

## 5.2. Settings of The Experiments

We conduct two different experiments. First, the eCMR and SI ontologies are directed aligned together with no use of background knowledge. Second, the experiment is conducted by the alignment of ontologies derived from the implementation guides of open standards with using LogiCO as an upper ontology. These choices will be further elaborated.

Using LogiCO will have some risks with respect to the experiment; it might not support the functionality of an implementation guide. To reduce this risk, an implementation guide of an open standard needs to be aligned as much as possible with LogiCO. Therefore, it is worthwhile to list the foundational concepts of LogiCO:

- Activity denotes some action and is relevant for the purpose of logistics, such as, for example, the activities of transport, storage, transshipment, loading, and unloading. Some activities are atomic and can be used to compose more complex activities.
- Event represents an occurrence of interest for the execution of a certain activity. In contrast to an activity, which denotes an action that is continuous in time, an event denotes an occurrence at a specific moment in time. For example, the departure of transport means from a location of origin and its arrival to the destination can be regarded, respectively, as starting and ending events for the transport activity.
- Actor represents organizations, authorities or individuals that offer or require activities and operate on resources related to these activities. An actor can have a Role, for example, customer and service provider, or shipper, consignee, forwarder, and carrier.
- Entity represents something that is used or exchanged during an activity. We specialize an Entity in a Spatial Entity, which represents tangible objects, such as an equipment or a person, and an Intangible Entity, which represents intangible objects, such as a modality, a characteristic or a dimension. We also define a Temporal Entity, which represents the start time, end time or time interval associated to activities and events. To this regard, since time is a basic (foundational) concept relevant for logistics, but common to other domains, we re-use the time ontology proposed by W3C (http://www.w3.org/TR/owl-time), instead of specifying our time ontology from scratch.

[5]LogiCO stands for **Logi**stics **C**ore **O**ntolgoy and is publicly available at http://ontology.tno.nl/logico

- Location represents the geographical area or geographical point used to define the place of origin and destination for entities and activities. Location can have different levels of granularity. Location can be coarse-grained for scheduling, since in long-term planning it is sufficient to specify approximately the place of origin and destination, such as, for example, the Netherlands or the port of Rotterdam.
- Moveable Resources are characterized by the capability of moving on their own or being contained in another entity for the purpose of movement, and Static Resources are used to host and/or handle moveable resources. An implementation guide has been constructed for an open standard representing document data for road transport. The open standard has been developed by UN/CEFACT for electronic waybills, with a specialization to the eCMR for road transport. The eCMR assigns one specific document type, the CMR, to a generic representation of data that can be stored by all types of transport documents. Thus, the core structure should as well be applicable for documents shared in other modalities. To conduct the experiments, the eCMR ontology discussed in previous section is used.

## 5.3. Experimental Results

The first experiment was the alignment of the two ontologies representing implementation guides of open standards, eCMR and SI. The alignment is not satisfactory. Only concepts representing common roles of organizations in the two ontologies can be aligned with each other, but not other concepts. In particular, the outcome of direct matching for SANOM was two correspondences containing the mapping of ***Consignee*** and ***Carrier*** that are identical in the two ontologies, AML detected one extra false positive by mapping ***ShippingInstruction*** to ***TransportInstruction***, LogMap mapped two identical object properties associated the two mapped classes, *isDeliveredTo* and *isCarriedBy*.

This mismatch of alignments is due to naming conventions that differ between the two ontologies in question. For instance, we used in the eCMR concept names derived from XML element names, like 'SupplyChainConsignment' and 'LogisticsPackage'. These concepts are not present in the other ontology. The concept 'SupplyChainConsignment' is also not expected to be part of a shipping instruction ontology, since the latter represents a consignment. In general, it is not common in supply and logistics standards to use a type of prefix 'SupplyChain' for naming concepts, which makes alignment only possible to those open standards that use the same prefix for naming. The same applies to 'LogisticsPackage'.

Furthermore, shipping instruction has additional roles, due to delivery conditions. Besides a consignor (which is equal to a shipper) and consignee, *notifies* will also be mentioned. A notify has to be informed when containers arrive at a port of discharge. Besides these differences in the naming of concepts, which will require a common data dictionary like the United Nations Trade Data Elements Directory (UNTDED), this naming difference might be solved by annotating LogiCO with terms used by other ontologies.

Another difference is that these open standards represent the transport services of enterprises like carriers. For instance, a shipping line is able to transport a container between the hinterland and a port and position a container for stuffing at the location of a shipper or only transport a container between two ports. The difference is known as carrier - and merchant haulage respectively and is, in fact, a combined service. This combined service is however not represented by an eCMR. However, the eCMR contains another modeling issue, namely that of modeling a shipment that can consist of more than one consignment. The shipment concept is used for data sharing between a customer and his carrier, the consignment concept for data sharing between a shipper and a forwarder.

A third difference is the representation of cargo. There are two different concepts used by these three ontologies, namely *LogisticsPackage* and *container*. One could argue that a container represents the actual cargo, but it also has packages stuffed inside. What is required besides agreement on the naming of concepts is the associations between those concepts, package and container.

We also use indirect matching to match eCMR and SI ontologies by the aid of LogiCO. Since LogiCO has also different naming from SI and eCMR, it is required to add some annotations to this ontology. In particular, we added the two extra annotations from the SI and eCMR ontology to LogiCO: we added *Shipper* from the SI ontology as synonym to *Consignor* and added *ShippingInstruction* from SI and *SupplyChainConsignment* from eCMR as synonyms to *Activity* in LogiCO. Table 1 shows the annotation of LogiCO concepts with those of SI and eCMR ontologies.

Table 1. The annotations made in LogiCO by using SI and eCMR terminologies.

| SI | LogiCO | eCMR |
|---|---|---|
| Shipper | Consignor | Consignor |
| ShippingInstruction | Activity | SupplyChainConsignment |

We experimented with three top alignment systems from the OAEI: SANOM, LogMap, and AML. These alignment systems had very promising outcome and producted identical results for the indirect matching. Figure 7 displays the mapping of eCMR concepts to those in SI that are obtained by indirect matching using the annotated LogiCO.

One of the reasons of having such outcome is that the object properties in two created ontologies are similar to each other. The identity of the object properties, such as *IsDroppedOffAT*, *IsDeliveredTo*, *IsPickedUpFrom*, as well as the annotations added to LogiCO increase the quality of the alignment by mapping the classes in the domains and ranges of the corresponding object properties.

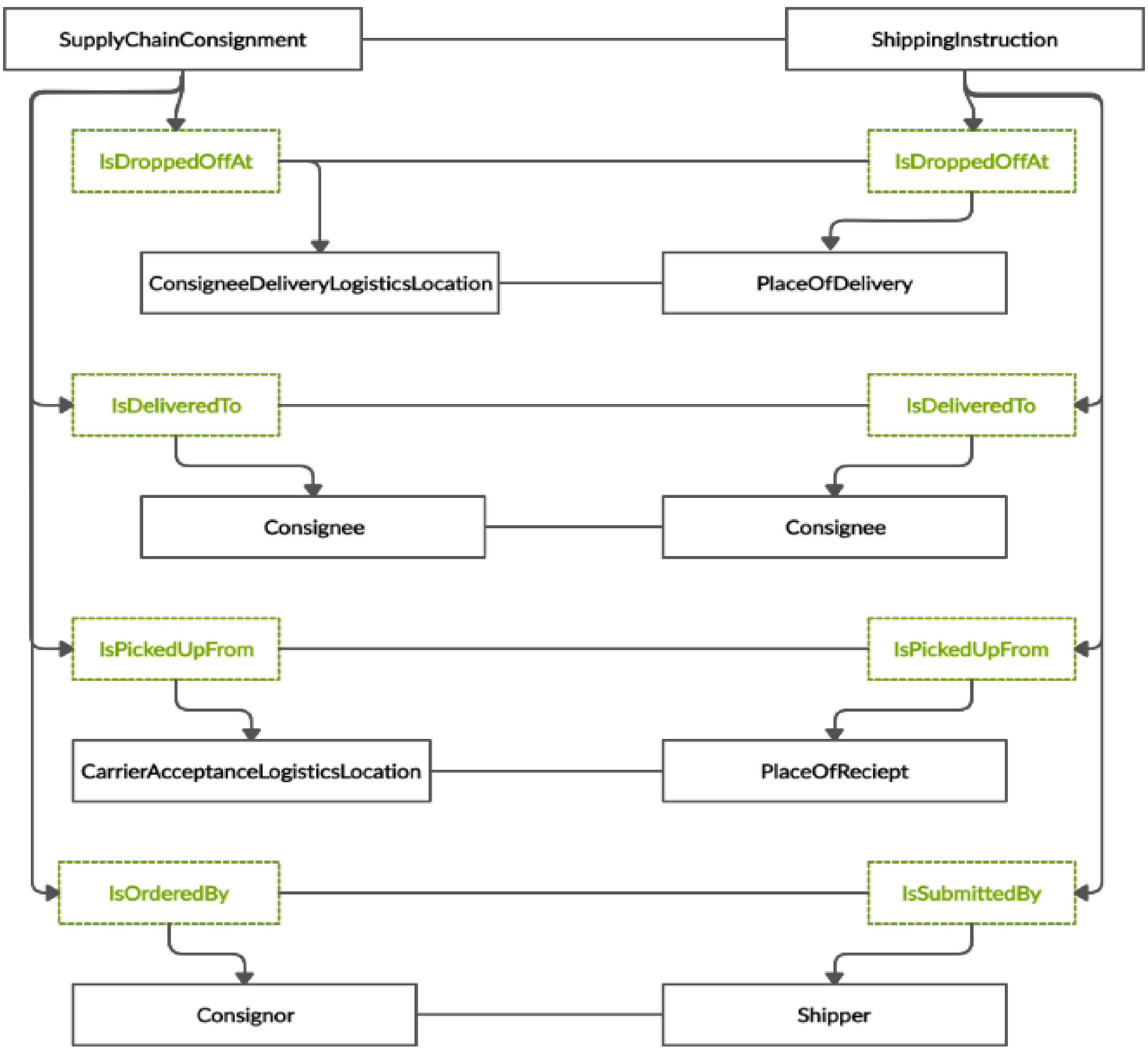


Figure 7. The alignment of eCMR to SI.

## 6. Conclusion

This paper addressed the main question of the applicability of ontology alignment to interoperability in logistics by means of an experiment. The experiment was the alignment of ontologies representing implementation guides of two open standards, eCMR and shipping instruction (SI), with and without the use of an upper ontology, called LogiCO. The experiments of direct alignment did not give satisfactory results due to differences in naming convention and systems modeled by the ontologies. A second experiment was performed by using indirect matching with the annotated LogiCO that resulted in a more acceptable alignment. The use of indirect matching is particularly useful in logistics since there are millions of enterprises. The indirect matching reduces the number of alignments from quadratic to linear with respect to the number of enterprises (or ontologies).

In view of the challenges encountered for alignment of implementation guides of open standards, it is safe to assume that alignment of (functional views of) database schemes represented as ontologies will even be more difficult. We cannot expect that ontologies derived from database schema use the same naming conventions and they will also have different structures, making the outcome of best ontology alignment systems, e.g., AML, SANOM, and LogMap, not acceptable and inappropriate for enabling interoperability in logistics. Note that these systems take advantages of several complex similarity metrics including string, linguistic, and structural, but were not able to detect enough correspondences. It is our expectation that ontology alignment will only improve if there is a common understanding of what needs to be represented by an upper ontology. The use of the upper ontology can enhance the outcome of matching systems if the upper ontology is properly annotated. However, the upper ontology has to be extended with knowledge of business service composition to address all possible standards. It means that we need to have a shared background knowledge for both standards and alignment development. In the latter case, the alignment systems may also have to be extended. A more complete upper ontology reduces the expert's efforts in annotating it and increases the performance of matching systems.

A thorough background knowledge can be created by using some existing ontologies in logistics like LogiCO and annotate it with proper names from different logistics models or standards. Such annotations require logistic expertise who is able to identify the related concepts in ontologies. An alternative approach is to create an integrated ontology by using ontology integration techniques based on the alignment of eCMR and SI, and then align new logistics models with the integrated ontology. Based on the alignment of the integrated ontology and the new logistics model, we can again use the ontology integration techniques to come up with a new integrated ontology. The alignment of each new model to the integrated ontology must be approved by a user so that the integrated ontology is reliable and does not contain redundant concepts. The creation of the integrated ontology is an iterative and augmentative approach that can finally result in a comprehensive upper ontology for logistics. While such an upper ontology can be exploited to enhance alignment outcome, it cannot be used for developing logistics IT systems since the integrated ontology gets more and more complex as the number of ontology alignment and integration increases. The complexity of such an integrated ontology makes it difficult for even logistics experts to comprehend or annotate it. The alternative solution is to create a comprehensive upper ontology for logistics that contain all the concepts and names in different standards. Such an ontology can be used by companies as well as it can help improve the alignment outcome.

## References


Giusti, R., Manerba, D., Bruno, G., and Tadei, R. (2019) 'Synchromodal logistics: An overview of critical success factors, enabling technologies, and open research issues', *Transportation Research Part E: Logistics and Transportation Review*, 129, 92-110.

Daniele, L. and Pires, L. F. (2013) 'An ontological approach to logistics', *Enterprise interoperability, research and applications in the service-oriented ecosystem, IWEI*, 13,199-213.

Erl, T. (1900) *Service-oriented architecture: concepts, technology, and design*, Chennai, India: Pearson Education India.

Euzenat, J., and Shvaiko, P. (2007) *Ontology matching*, Heidelberg, Germany: Springer.

Glöckner, M., and Ludwig, A. (2017) 'Ontological structuring of logistics services', *International Conference on Web Intelligence*, Leipzig, Germany, 146-153.

Dijkman, R., Hofstetter, J., and Koehler, J. (2011) *Business Process Model and Notation*, Heidelberg, Germany: Springer.

Hofman, W. (2019) Toward large-scale logistics interoperability based on an analysis of available open standards, In *Enterprise Interoperability VIII* (pp. 249-261), Heidelberg, Germany: Springer.

Hofman, W. J. (2019) 'A Methodological Approach for Development and Deployment of Data Sharing in Complex Organizational Supply and Logistics Networks with Blockchain Technology', *IFAC-PapersOnLine*, 52(3), 55-60.

ISO. (1988) Electronic data interchange for administration, commerce and transport (edifact)-application level syntax rules.

Lohmann, S., Negru, S., Haag, F., and Ertl, T. (2016) 'Visualizing ontologies with VOWL', *Semantic Web*, 7(4), 399-419.

Mascardi, V., Cordì, V., and Rosso, P. (2007) 'A Comparison of Upper Ontologies', *Woa*, Genova, Italy, 55-64.

Satta, G., Parola, F., and Lee, S. W. (2011) 'The EU-27 logistics industry: structure and trends of major subsectors and national markets', *Kmi International Journal of Maritime Affairs and Fisheries*, 3(1), 1-32.

Schönberger, A., Wilms, C., and Wirtz, G. (2010) 'A requirements analysis of Business-to-Business integration', Available at https://core.ac.uk/download/pdf/144484319.pdf

E. Union. (2017) *New European interoperability framework promoting seamless services and data flows for European public administration*, Luxembourg, Luxembourg: Publication Office of the European Union.

Wang, W., Tolk, A., and Wang, W. (2009) 'The levels of conceptual interoperability model: applying systems engineering principles to M&S', *arXiv preprint arXiv:0908.0191*.

Töyli, J., Lorentz, H., Ojala, L., Wieland, A., and Wallenburg, C. M. (2013) 'The influence of relational competencies on supply chain resilience: a relational view', *International Journal of Physical Distribution & Logistics Management*, 43(4), 300-320.