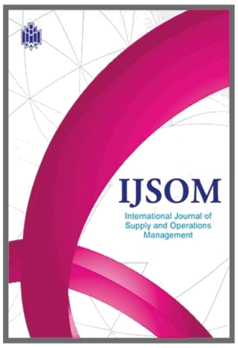
International Journal of Supply and Operations Management

IJSOM

2022, Volume 10, Issue 4, pp. 456-484
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com



An Ontological Approach to Model Outbound Logistics based on Internet of Things (OLP-IOT)

Saloua Aoulad Allouch [a*], Khalid Amechnoue [a] and Iman Achatbi [a]

[a] *National School of Applied Sciences Tangier, Morocco*

Abstract

Currently, the performance of outbound logistics processes is an important element for companies that would like to increase the level of customer satisfaction and improve the visibility of the supply chain, thus guaranteeing the quality and safety of products. On the other hand, the concept of smart logistics has been proposed as a technological solution that aims to improve performance, security and traceability in the logistics of companies. A crucial element to achieve this goal is to benefit from the emergence of the Internet of Things (IOT) and related technologies. Indeed, IOT streamlines the logistics process and improves its efficiency. However, to track and trace a product's life cycle, its physical state, associated activities, and involved objects, a large amount of heterogeneous data will be generated from various sources, especially from sensors and RFID (Radio Frequency Identification) tags. Moreover, the observations produced by these sources are made available with heterogeneous vocabularies and data formats. This heterogeneity creates interoperability problems and prevents the adoption of generic solutions on a global scale which make it difficult to reuse data for other purposes and share them among different stakeholders. To address this challenge, we propose in this work an approach based on semantic modeling, using the semantic web and ontologies, to improve the interoperability and knowledge sharing of different phenomena in the outbound logistics domain. In this sense, we have designed and developed an OLP-IOT ontological approach adapted to logistics data and offering semantic enrichment of IOT data. This approach allows the sharing of sensor observations, the identification of products and logistics objects involved as well as the contextualization of data and the reuse of processed knowledge and information. The ontology was developed using the Neon methodology, which emphasizes reuse and modularization. This explicit knowledge is then used to develop a reasoning system to guide the logistic expert for an incremental and semi-automatic construction of a software solution to an instantaneous problem.

Keywords: Supply chain visibility, ontology, semantic web, semantic interoperability, Neon methodology, reasoning.

1. Introduction

Actually, Supply Chain (SC) faces increasingly complex and rapid changes in markets and customer's needs (Butner, 2010), in addition to new risks and challenges created in such ever-changing environments (Rejeb et al., 2020). These risks are essentially about visibility, reliability and relationships (Christopher, 2016). In order to be competitive, companies must develop a new and a smarter SC that can enhance collaboration between stakeholders, improve the efficiency of different operations and increase the level of visibility in the whole SC.

Logistics, as a key part of the SC process, must adapt to these new strategies to meet the new demands of globalization and market changes. We can define logistics as a set of processes focused on managing and controlling the transportation and storage of materials, goods, and related information flows from the point of origin to the point of consumption (Christopher & Holweg, 2011). Logistics includes inbound and outbound logistics. Inbound logistics refers to the purchase, procurement, and movement of raw materials, parts, and/or finished inventory from the supplier to manufacturing plants or warehouses. On the other hand, outbound logistics is the process that includes the storage, packaging and transportation of finished goods and related information from the end of production to the end users.

---

*Corresponding author email address: saloua.aouladallouch@gmail.com
DOI: 10.22034/IJSOM.2023.109230.2270

Currently, the performance of outbound logistics processes is an important determinant for companies that would to increase the level of customer satisfaction and improve the visibility of SC, thus ensuring the quality and safety of products, especially sensitive products. On the other hand, new technologies, such as the Internet of Things (IOT) (Tamrakar et al., 2022), are significantly streamlining the logistics process and improving its efficiency (Verdouw et al., 2016a). According to the report published by (Martin Placek &, 2022), the global industrial IOT market has been valued at more than US$263 billion in 2021, and is estimated to grow in size in the coming years, reaching some US$1.11 trillion by 2028. Therefore, the advent of IOT brings a new approach to collecting, transferring, storing and sharing information across the SC. To be competitive, companies need to develop a new, smarter SC that can improve collaboration between stakeholders, efficiency of different operations, and the level of visibility throughout the chain (Ding et al., 2021a). Therefore, there is no doubt that developing smarter approaches to improve logistics efficiency and reduce its costs, both in academia and industry, is a timely and important topic today (Ahmed et al., 2021). Figure 1 illustrates the main industrial applications of IOT. According to (Ding et al., 2021b), these applications consist of the perishable food industry, agriculture, chemical industry, e-commerce, pharmaceutical industry, fishing industry, and tobacco industry. The cold chain logistics of the perishable food, agriculture and fishing industry is dominant and accounts for 61% of the studies. It is followed by the chemical industry which constitutes 17% of the studies.

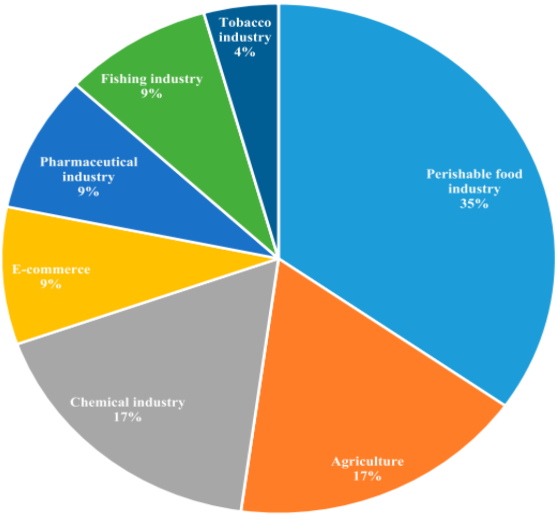


Figure 1: Major industrial applications (Ding et al., 2021b)

The logistics domain may have a relatively large and varied information sources, likewise, the number of sensors, actuators and different devices that join IOT systems is increasing, producing a massive real-time data flow. Moreover, the variety of these data in terms of types, contents and sizes triggers interoperability and communication challenges between actors and IT systems. On the other hand, agri-food, pharmaceutical (Helwan University & Ouf, 2021) and perishable supply chains (Jachimczyk et al., 2021) are highly complex and dynamic networks that require intensive information exchange (Tran-Dang et al., 2020). Reliable information must be shared in a timely manner throughout the chain and real-time visibility of material and product flows must be provided (Sun et al., 2020). Subsequently, savvy companies need to react quickly to changes, taking into account varying quality parameters, organizational conditions, and market segment requirements. To cope with these challenges, companies must use standards and common semantics to describe and interpret concepts involved and data generated from heterogeneous entities and IOT devices, in addition to understanding activities and events influencing the product's life cycle (Symeonaki et al., 2022a). Accordingly, the use of ontologies can improve addressing of phenomena occurring in the outbound logistics and facilitate communication between stakeholders and machines (Sure et al., 2009).

Numerous ontologies for SC and IOT have been developed (Fraga et al., 2020), (Wang et al., 2019). However, most of them focus on manufacturing and production processes (Jaskó et al., 2020), others, do not apply IOT ontologies to the logistic domain (Yang et al., 2019) or they focus on specific components of IOT technologies; we can argue that the use of both IOT and ontologies in logistic modelization is still in its early stage. To this end, our goal is to develop an ontological approach to model the integration of IOT technologies in the outbound logistics and use reasoning functions to address heterogeneity problems, data representation, and enhance visibility and collaboration in the SC.

The remainder of this paper is structured as follows: Section 2 presents a literature review of research related to IOT ontologies, logistic ontologies and their intersection. Section 3 gives a brief review of existing methodology to construct ontologies and describes methodology adopted to develop our ontological approach. Section 4 introduces OLP-IOT ontology and the process followed for its design and development. The formalization of rules used in our ontologies to infer new information and the evaluation of the proposed ontological approach are presented in Section 5. Section 6 and

7 present an application of the proposed model and the managerial implication respectively. Future work and conclusion are given in Section 8.

2. Related works

To ensure our goal, we started with a comprehensive literature review that gives a vision of what is actually developed by previous research in this area and what gaps we will address. To this end, we need to examine the most important contributions in the three topics of our research (ontologies, IOT and logistics).

Much study has been focused on the use of RFID and IOT in smart design, manufacturing and production control; furthermore, they address a special application area as food and healthcare industries. As an example, (Verdouw et al., 2016b) proposed an information architecture system based on the IOT to implement the concept of virtualization in the food SC. This architecture has been applied and validated by a case study of a fish supply chain. Also, (Choy et al., 2017) have proposed RFID-SAS: RFIDs based storage assignment system; a rule-based system incorporating RFID and fuzzy logic to resolve the storage location assignment problem in a warehouse, in order to enhance efficiency in order picking. Authors in (Yuen et al., 2018) proposed the IOT Knowledge Management System (IOLMS). The objective of this system is to guarantee the real-time environment and product monitoring in the context of outbound logistic processes. The system does not use ontologies in the process of modeling.

Various literature reviews that concern application of the IOT and RFID in SC appeared in the last decade. For more details, the reader can refer to many of these works such as: (Ben-Daya et al., 2019) which presents a review of the IOT and its technologies in different sectors of the SC. For (Ding et al., 2021b), they provide a methodical summary based on publication between 2008 and 2019; it identifies the latest research and applications of smart logistics based on IOT, including warehousing, smart freight transport and delivery. Authors in (Tran-Dang et al., 2020) aim to examine the state of the art of IOT applications in the logistics industry.

Several definitions were given in the last decades to ontologies, the one that characterize the principal aim of ontology is given by Gruber; he defines ontology as an explicit specification of a shared conceptualization (Gruber, 1993). Ontologies are developed to provide a model that is machine processable and allows a common understanding and a shared vocabulary between different software systems and actors (Klein et al., 2001).

In this section, we describe some of available ontologies in the domain of SC and IOT with a special focus on those that have a direct relation with our proposed contribution.

2.1. Supply chain ontologies

Several research works have already been done in SC ontologies to discuss the problem of semantic interoperability between stakeholders. In this context, ontology for the logistic process based on semantic models has been proposed in (Lian et al., 2007) in which authors use a bottom-up approach to define situations of products and events triggered by these situations. In the same context, (Park et al., 2008) have extended this work by introducing an ontological approach to integrate distributed logistic information into EPC network to enhance collaboration between logistics actors and share a common vocabulary. In addition to enable reusability and extensibility of the proposed ontology, authors define a logistic ontology based on sub-ontologies of the three main concepts extracted to represent the semantic integration of distributed logistic information: state, activity and event. However, these works still lack the integration of logistics resources involved in outbound logistics processes, in addition to different component of IOT to provide observations and measurements.

From the point of view of product, (Tursi et al., 2009) have proposed a bottom-up approach based on a product ontology to provide a common information model that include most semantic concepts and rules related to product. However, no interest was given to the logistics process or other IOT concepts in the product ontology. In the same perspective, (Lu et al., 2013) have proposed a product centric SC ontology framework to enhance interoperability between enterprises involved in the SC interactions. The ontology proposed is based on the alignment of SCOR (Zhou et al., 2011) model and ONTO-PDM ontology previously defined by (Panetto et al., 2012). (Daniele & Pires, 2014) described the basic ontology for logistics focusing on the physical resource concept. A limited taxonomy of resources, their structures and some axioms in reference to the relationships between resources were proposed.

Over the last decade, several ontologies have been developed for a specific industrial domain such as aviation (Keller, 2016), chemical engineering (Vinoth & Sankar, 2016), energy (Santos et al., 2018), traceability in meat supply chain (Pizzuti et al., 2017), (Dooley et al., 2018), pharmaceutical supply chain management systems (Helwan University & Ouf, 2021). Other ontologies have been used for a specific manufacturing process, such as process engineering (Hooi et al., 2012), customer feedback analysis (Daly et al., 2015), organizational management (Grangel-Gonzalez et al., 2016), product development (Zhang et al., 2017), and production planning (Kourtis et al., 2019).

(Cheng et al., 2016) provided a model of the production line using a combination of five ontologies: the device ontology, the process ontology, the parameter ontology, the product ontology, and the core ontology. For (Engel et al., 2018), they proposed a three-layer ontology for batch plants. The application layer contains the operations; the domain layer contains the architecture, while the top layer refers to a higher ontological model describing the general characteristics and relationships of the system. On the other hand, (Zheng et al., 2020) provide domain-level ontology as common information for logistics operations in the construction industry to improve the efficiency and transparency of logistics information management. (Cao et al., 2020) construct a "multi-view risk ontology" that can produce a consistent common understanding of risks, their origins, and their impacts in global fresh produce supply chains.

2.2. IOT ontologies

Several IOT ontologies have been introduced to solve heterogeneity problems and enhance semantic interoperability. As an example, **Semantic Sensor Network (SSN**) (Compton et al., 2012) a standard ontology proposed by W3C Incubator Group and recommended by Linked Open Vocabulary (**LOV**)[*]. It solves heterogeneity problems by semantically describing sensors, their capabilities and observation phenomena. However, SSN focuses exclusively on sensing activities and does not cover other important aspects of the IOT such as actuation. Moreover, it lacks concepts to define features of interest, unit of measurement and other spatial and temporal aspects. As a complement of SSN, Semantic Actuator Network (**SAN)** [†]has been proposed to semantically describe actuators and all concepts needed to include actuating services, also, SAN imports some concepts from SSN ontology. SAN and SSN have been used as a base to constitute **IOT-O** (Seydoux et al., 2015); a core domain IOT ontology following best practices to describe knowledge about IOT systems and applications in order to support heterogeneity. It also used concepts from **DUL**[‡] and **QUDT**[§].

On the other hand, many contributions have been done to reuse and combine existing ontologies to provide new ontologies for specific platforms in the IOT domain. In this context, **IOT-Lite** (Bermudez-Edo et al., 2017) has been introduced as a core lightweight ontology that instantiates SSN ontology and describes only the most used terms in the IOT domain by a lightweight semantics allowing interoperability. Furthermore, IOT-Lite can be used as a reusable model aligned with other semantic modules to provide other new typical applications. In the same context, **FIESTA- IOT** (Agarwal et al., 2016) ontology has been proposed based on Noy et al. methodology for reusing and combining existing ontologies in a one unified ontology. It includes a set of core concepts generated from well-known ontologies and taxonomies such as SSN, IOT-Lite, **M3-Lite**[**], **Time**[††] and **DULL**. The main objective of this ontology is the integration and update of these concepts to ensure a semantic interoperability between diverse platforms and test beds, in addition to semantic description of IOT resources (i.e. sensor, tags, etc.) and related observations. After rethinking the Semantic Sensor Ontology (SSNX) and overcoming its shortcomings, the Sensor, Observation, Sample, and Actuator ontology (**SOSA**) (Janowicz et al., 2019) was introduced as a core structure. It is based on changes in scope and audience to address important aspects of IOT such as real-time data collection, measurement units, actuation and sampling activities. SOSA presents the interactions between the physical and digital world by modeling activities of observing, actuating and sampling, in addition to devices and relations between them. Accordingly, SOSA has been introduced as a core lightweight for a new version of **SSN** [‡‡] to address challenges encountered by the original SSN. In contrast to the old SSN, the new SSN contains several sub-ontologies and modules that differ in the scope of coverage and permits use of required concepts for specific implementation or applications. The authors in (Rahman & Hussain, 2018) proposed a lightweight ontology (LiO-IOT) which considers sensors, actuators and RFID as IOT concepts. For (Elsaleh et al., 2019),(Elsaleh et al., 2020) they present a lightweight semantic model for annotating streaming data. The suggested model extends existing ontologies as SSN and SOSA. (Benkhaled et al., 2022) propose single cross-domain ontology (CDOnto) for semantic interoperability between different IOT domains. The ontology brings together all data related to devices and their measurements under one roof. It is considered a generic model which can be extended by domain-specific ontologies. In the same context, (Swar et al., 2022) suggest a shared and a unified ontology schema to address IOT integration problems. It is based on the combination of several ontologies as: SOSA, SSN, GEO (Zhong et al., 2017), QU, and IOT-Lite.

2.3. Research gap

The literature is recognizing more and more the importance of IOT in increasing the visibility and accuracy of the SC. In the same context, studies assert that using ontologies is essential for improving interoperability and collaboration between various stakeholders. However, current research efforts overlook an essential aspect, namely, the development of ontology which can include both logistics and IOT characteristics. In other word, how IOT technologies are incorporated to ensure visibility, traceability and sharing of information throughout outbound logistic processes. Accordingly, our proposed

---

[*] http://lov.okfn.org/dataset/lov/

[†] https://www.irit.fr/recherches/MELODI/ontologies/SAN

[‡] http://www.loa.istc.cnr.it/ontologies/DUL.owl#

[§] http://qudt.org/schema/qudt#

[**] http://purl.org/IOT/vocab/m3-lite#

[††] http://www.w3.org/2006/time#

[‡‡] https://www.w3.org/ns/ssn/

model is distinguished from the existing works in such criteria. On the one hand, most ontologies which include IOT to ensure SC visibility do not provide a real description of these technologies and how they are integrated into the SC (Jachimczyk et al., 2021) (Cao et al., 2020). Moreover, they are limited to a static description of the products and the logistic processes without considering the dynamic aspect which takes into account the requirements and the particularity of each type of product as well as the environmental conditions (Jachimczyk et al., 2021). However, our ontological approach takes into account the description of IOT technologies such as sensors and actuators, in addition to their interaction with the equipment and the environment of the outbound logistics to guarantee the interoperability of both physical and informational flow and to react in case of abnormal situations. On the other hand, several ontologies proposed are restricted to a specific type of product including for example agricultural sector (Symeonaki et al., 2022), pharmaceutical supply chain (Helwan University & Ouf, 2021) and dairy supply chain (Jachimczyk et al., 2021). To fill this gap, our work is intended to deal with outbound logistics by implementing an ontology considering the particularity of all types of products.

Research on this domain can be filled in by this study, and thus environmental conditions and different requirements needed to guarantee a good quality and safety of products will be considered by using our ontological approach and by taking into account all the data generated, interpreted and inferred.

3. Research methodology

Several methodologies have been proposed by researchers to assist the development and reusing of ontologies. In our ontology development process, we have opted to use **Neon methodology** (Suárez-Figueroa et al., 2012a)**;** which is a scenario-based methodology, conceived for building ontology networks and supports different aspects of ontology network development. It covers various scenarios of ontology development; from scratch and by reusing and restructuring existing ontologies. This methodology provides a set of nine scenarios involving different methods and techniques to enhance reusing, engineering of knowledge resources (ontological and non-ontological), in addition to restructuring and merging processes.

These scenarios are decomposed in different processes and activities; a set of methodological guidelines is described for each process and activity. Neon methodology presents two ontology network life cycle models to organize activities and describe how scenarios will be applied: the waterfall model and the iterative-incremental model.

Scenario 1: From specification to implementation. This scenario aims to develop network ontology from scratch by using knowledge about the domain.

Scenario 2: Reusing and reengineering non-ontological resources. It aims to build ontologies by using non-ontological resources (thesauri, lexicons and classification schemes). This, by following some activities which involve searching non-ontological resources; evaluating the set of non-ontological resources and finally selecting the most appropriate resources.

Scenario 3: Reusing ontological resources. It is used when ontology developers aim to take advantage of existing ontological resources that are related with their domain of interest. To reuse existing ontologies some activities must be followed: Ontology research, ontology assessment, ontology comparison, ontology selection and ontology integration.

Scenario 4: Reusing and re-engineering ontological resources. Ontology developers reuse ontological resources to build network ontology, and this after modifying and performing them to be more suitable and useful for the purpose. Activities involved in this scenario include the process of reusing ontological resource, in addition to the process of ontological resource reengineering which is composed of the following activities: *ontological resource reverse engineering*, *ontological resource restructuring*, and *ontological resource forward engineering*.

Scenario 5: Reusing and merging ontological resources. It is applied when ontology developers need to create a new ontological resource from two or more ontological resources in the same domain. It involves activities described in scenario 3 to reuse ontological resources.

Scenario 6: Reusing, merging and reengineering ontological resources. This scenario is similar to scenario 5, the difference is that merged resources must be modified before being used to meet the intended purpose.

Scenario 7: Reusing Ontology Design Patterns (ODPs). Ontology developers' reuse existing ontology design patterns to build the ontology network, this process includes such activities: ODP search, ODP selection, ODP adaptation and ODP integration.

Scenario 8: Restructuring ontological resources. In this scenario, ontology developers restructure ontological resources to be correctly used for the intended purpose. This scenario can involve different activities as modularization, pruning, extending and/ or specializing.

Scenario 9: Localizing ontological resources. This scenario assures the transformation of all the terms of ontology into another natural language.

In this work, the Neon methodology was chosen as the most suitable methodology for our purpose. Indeed, the construction of our ontological approach OLP-IOT is based on the combination of two domains of interest: logistics and IOT. To model these two domains, we need knowledge from ontological and non-ontological resources. This knowledge needs to be reorganized, restructured and merged to be appropriate for our research objective. In this context, the choice of Neon is mainly justified because it includes a variety of scenarios that describe all the activities needed in our context.

4. The OLP-IOT ontological approach

In this section, we present how the OLP-IOT ontological approach was built, and the main components involved in solving the problem of heterogeneity and visibility in outbound logistics processes. As we have opted to use the Six-Phase + Merging Phase Waterfall Ontology Network Life Cycle Model (Suárez-Figueroa et al., 2012a) in the development of our ontology, the building process starts by initiation phase in which ontology requirements specification are detailed. Then, comes the process of research, assessment and selection of most suitable knowledge resources for the reusing phase (scenario 2 and 3). These resources will be merged and reengineered in merging and reengineering phases to be used in the design phase (scenario 6 and 8). The details of different phases of this model are presented in sections below.

4.1. Ontology Requirement Specification

Currently, various devices are involved to enhance the traceability and visibility of outbound logistics using IOT technologies. The heterogeneity of these devices and the concepts included influences the interpretation of the produced information. Consequently, it impacts the traceability of the logistic objects and the ability to react in case of abnormal situations. In fact, there is no common standard to model how IOT technologies are applied and used in logistics and how the shared information should be interpreted in order to track and trace the life cycle of product or involved logistics units. To address these shortcomings, OLP-IOT ontology will provide a shared and common reference model capable of describing all relevant components and knowledge to improve traceability and interoperability of outbound logistics flows between stakeholders. Outbound logistics is a core part of SC where stakeholders (retailer, shipper, wholesaler, carrier, distributors...) collaborate to ensure safety and accuracy in the storage and transportation of the final product. This can be achieved by semantic and comprehensible sharing of product information and all activities involved in this process. Product related information consists of regrouping details of: internal components, physical attributes and requirements to transport product in a good condition and other observable properties that describe the transition of the product through the SC. To detect these specific features, the IOT is mainly adopted and involves different technologies as sensors, actuators and tags; the use of these technologies allows actors to have the right information, at the right time for dynamic control and real-time processing. This is especially in the case of an emergency situation as exceeding the maximum temperature allowed for the transportation of product.
To highlight the functional requirements that must be respected in the elaboration of OLP-IOT ontology, a set of competency questions was used. In the following list, we illustrate some competency question used in the elaboration of our ontology:

CQ1: What is the percentage of humidity in the container C?
CQ2: Which temperature and humidity sensors were used for product P?
CQ3: What sensors were used in warehouse X and what properties do they measure?
CQ4: What are the maximum and minimum temperatures required for all products stored in a location X?
CQ5: What are the temperature values observed in the warehouse in which a product P is stored during a period [T1, T2]?

The OLP-IOT ontology elaboration process was implemented on Protégé[*] software; which provides an environment for users to edit and create ontologies with various facilities. For our ontology development, we have opted to use Ontology Web Language (OWL)[†] as a representation language. In addition to the capacity of describing classes and properties as other ontology languages; OWL integrates tools to express constraints and comparing properties and classes as: identity, equivalence, symmetry, cardinality, disjunction, inverse and others.

4.2. Selection and reuse of resources

OLP-IOT is based on the combination of two domains of interest: outbound logistics and IOT. Outbound logistics is represented by important concepts involved in this domain, relationship between them and description of the phenomena that can occur. The IOT domain, represented by IOT technologies and aspects used to permit traceability and visibility of all items and equipment involved in the outbound logistics process.

---

[*] https://protege.stanford.edu/
[†] http://www.w3.org/TR/owl-ref/

To ensure the goal of our contribution, we have opted to start our development by searching for ontological and non-ontological resources that can be reused or restructured for use in each of these domains. The following subsections describe all the resources used in our ontology approach.

4.2.1. Reusing and restructuring of IOT resources

Based on the richness of Linked Open Vocabularies (LOV)* database and the high quality catalog of reusable vocabularies that it offers, we studied a variety of IOT ontologies and vocabularies it contains. We found that IOT ontologies are general or encompass concepts related to different domains such as environment, agriculture, smart home…etc. In this context, we decided to use SOSA and M3-Lite to capture the main IOT concepts and relationships which describe activities of tracing, tracking in the context of outbound logistics processes.

We have opted to use **SOSA** as it is a core and domain independent ontology which describes common and generic concepts used in the majority of IOT applications. This ontology is generic and can be extended with application domain concepts such as actions and activities related to the logistics domain. Therefore, SOSA ontology has been imported, pruned and extended with additional classes, properties and relationships.

In order to present the requirements of the domain, SOSA needs to be extended and specialized vertically. Thus, to integrate all aspects of outbound logistics, we need to describe the different types of sensors, actuators and measurement required. Furthermore, applications focusing on the monitoring and control of products and their environment are essentially based on data generated by these devices. In this context, we have decided to use the **M3-Lite** taxonomy to complete SOSA as it lists different types of sensors and the typical domain in which they can be used. In addition, it includes the **RFID** concept, different units of measurement and properties that can be used to extend the top-level IOT ontology extracted from SOSA.

A pruning process was also performed for the M3-Lite taxonomy; we retained only those concepts relevant to our purpose. These concepts describe especially a classification of sensors, actuators and properties in relations with logistics processes. Similarly, we also followed scenario 2 of the Neon methodology, where we relied on the taxonomy presented in the work of (Rozsa et al., 2016) which is applied to categorize the main sensors used to create IOT applications.

4.2.2. Reusing and restructuring logistics resources

We considered using the ontologies LogiCo, LogiServ and the model proposed by (Lian et al., 2007) as well as the standard provided by (SCG) which serves as a basis to determine the product type. The main objective of the model proposed by Lian is to present the state and context of objects (Device, Product) and their environment. For equipment, they represent just vehicles, so no interest was given to different logistics equipment used during all outbound logistics. Similarly, the authors describe concepts of the logistics process and a set of logistic actions and triggered events. However, this model failed to capture the important properties and relationships that can describe the actual transition of products and logistics units, especially in the packing, loading, and transportation processes. This ontology is not available online in a machine-readable format and we cannot reuse it directly. For this reason, it will be considered as a non-ontological resource (Neon scenario 2) and only required information will be extracted from this resource to be used in the development of our ontology.

Moreover, due to the insufficiency of the model proposed by (Lian et al., 2007), ontology modules produced by **TNO** †: **LogiCo**‡ and **LogiServ**§**;** were used to give an overview of the structure and the core concepts used in the logistics domain, especially concerning products, moveable and static resources, in addition to different transport modes and means used to achieve this goal (Daniele & Pires, 2014). More interest was given also to the physical attributes and parameters of the logistics objects. As only some concepts of LogiCo will be reused in our development process and to perform ontology maintenance and performance, LogiCo ontology will not be imported in our design. We will focus on interconnecting our ontology with principal information of LogiCo and LogiServ required in our development process.

4.3. The conception and formalization of ontological approach

In order to keep the ontology extensible and eventually offer a high-level structure, a modular and extensible approach was used. The adoption of modularization as a basis for building OLP-IOT consists in dividing the knowledge of each domain into modules, containing different types of knowledge. According to (d'Aquin et al., 2007), modularization represents a way of structuring ontologies and allows the design of large ontologies, based on the combination of autonomous, independent and reusable knowledge components. Extensibility is another key feature of the ontology, so

---

[*] https://lov.linkeddata.es/dataset/lov/

[†] https://ontology.tno.nl/

[‡] https://ontology.tno.nl/logico/

[§] https://ontology.tno.nl/logiserv/

that new concepts and modules can be easily added or removed while reducing time and effort. In this work, we consider these components as ontologies, called "modules", and that the resulting ontology of this combination is a "modular ontology". As a result, we have defined three ontologies: IOT4Log, OLP-Onto and OLP-IOT.

4.3.1. IOT4Log ontology

The IOT4Log ontology is proposed to model IOT concepts which are important to enable product traceability and status tracking in the outbound logistics domain. It is an important module that can be used in other work related to logistics. The main role of this ontology is to describe different properties observed on which we can act automatically, as well as identifying technologies (sensor, actuators) used to permit these observations and measurements in both physical and informational flow of the outbound logistic. Therefore, to ensure a formal description of the shared information between IOT devices and logistics stakeholders, it is necessary to have a hierarchy of sensors and actuators used in the context of the outbound logistics. Moreover, the relationships between these different components as well as the physical properties that can be used to track and identify the state of a product or its environment are necessary to define a formal description of these devices. Figure 2 shows a general overview of the classes and relations used in IOT4Log, it indicates the prefix of each class used to distinguish between concepts defined in IOT4Log (prefix: *IOT4Log*) or reused from SOSA and M3-Lite ontologies (prefix: *SOSA*, *M3-Lite*).

The observation module is the core of the IOT4Log ontology. Essentially based on the SOSA ontology and the M3-Lite taxonomy, it allows the description of an indicator as the result of an observation.

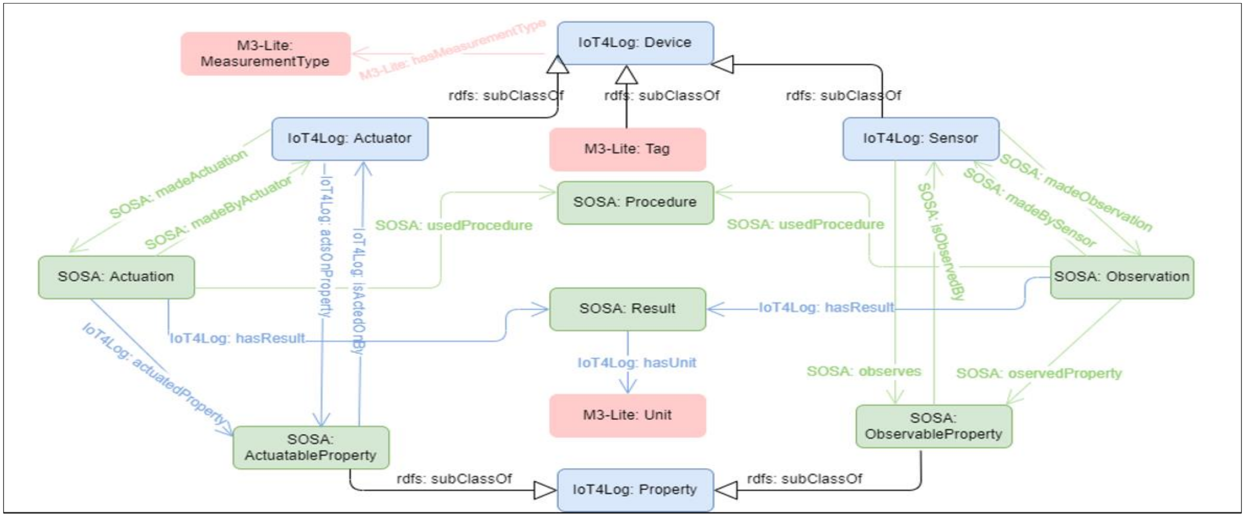


Figure 2: Global architecture of the IOT4Log ontology

Observation (i.e., monitoring or detection) of an object or place is the primary motivation for capturing the environment. The main objective of this module is to enrich the heterogeneous sensor data with a formal semantic representation containing more details about the operation performed by the sensor. The latter is the key element of this operation; its main function is to detect environmental measurements and send a readable message to the network.

The term Observation is used to describe both an observed property or feature of interest and a value assigned to that property by a particular sensor. Therefore, the result of the observation is the output of a sensor. The concept of observation is defined in the SOSA ontology and reused in the IOT4Log ontology.

In the following, more details of each concept included in our work will be provided:

SOSA: Observation class provides the structure for representing a single observation. Several properties are defined for this class instance, some of which are summarized below and illustrated in Table 1.

SOSA: observedProperty points to any property observed by a particular sensor, e.g. soil moisture, temperature, air humidity, etc.

SOSA: madeBySensor: links the instance of an observation to the sensor that made it, for example a humidity sensor.

IOT4Log: hasResult: links an instance of *SOSA: Observation* to its result (*SOSA: Result*) which is the output of a sensor.

SOSA: usedProcedure describes the relation between the observation and the procedure used.

SOSA: Procedure class is reused from the SOSA ontology. A procedure is a workflow, protocol, plan, algorithm or computational method that specifies how to perform an observation or change the state of the world via an actuator. Therefore, procedures are also crucial to fostering semantic interoperability, as procedures related to observation activities are typically a record of how those activities are/were performed (Janowicz et al., 2018).

IOT4Log: Device is used as a main component of IOT4Log ontology, it represents in our context a physical piece of technology which is a superclass of materials used to observe, identify or actuate on a resource. *IOT4Log: Sensor*, *IOT4Log: Actuator* and *M3-Lite: TagDevice* are introduced as sub-classes of Device class.

RFID tags are embedded in products, logistic loading units and all equipment used in outbound logistics, to allow their identification and traceability throughout the processes and activities. To model these tags, we have reused the ***M3-Lite class: TagDevice***. This component is essential to ensure both: observation, by following the products and logistic objects concerned to capture measurements, and also to ensure actuation, by identifying the product or logistic object on which we want to make actions or changes.

To represent the various sensors in the IOT logistics infrastructure, the ***IOT4Log: Sensor*** class is used. Based on the taxonomic classification introduced by (Rozsa et al., 2016) and our requirements for outbound logistics processes, we specialized the *IOT4Log: Sensor* class with three main types: *IOT4Log: PositionSensors*, *IOT4Log: EnvironmentSensor* and *IOT4Log:MassMeasurementSensors.*

Each of these types was specialized to describe more specific types of sensors, this by using some subclasses of M3-Lite: SensingDevice with other types of sensors that we have added by leveraging literature and documents provided by logistics experts, as illustrated in Figure 3 and described in the text below:

IOT4Log: PositionSensors; measures the position of a logistics object as (Proximity sensor, presence detectors, accelerometer, GPS sensor).

IOT4Log: EnvironmentSensors; is used to measure environmental properties of outbound logistics (*AirPoullutatSensor, HumiditySensor, LightSensor, GasDetector, SmokeDetector, Thermometer…*).

IOT4Log: MassMeasurementSensors; measures a property of a logistics object or a physical interaction force with a logistic object (*PressureSensor*, *WeightSensor*, *PalletSensor*, *VolumeSensor*, and *LoadSensor*). The specialization of each type of sensor was represented using *rdfs: subClassOf* relations.

The *IOT4Log: Sensor* class has three main object properties as presented in Figure 2: *SOSA:madeObservation* describes the instance of the observation made by this sensor, *SOSA:observes* describes the property observed by the sensor, e.g. air humidity, air temperature, etc. The *M3-Lite: hasMeasurementType* property describes the type of measurements provided by the sensor instance concerned.

SOSA:ObservableProperty represents the property of the logistic object or its environment that will be observed by a particular type of sensor. Since this concept in the SOSA ontology remains generic and does not present the specific types of sensors used in the context of logistics, we have specialized it by **QuantityKind** subclasses of the M3-Lite pruning ontology, in addition to other properties that we have added according to the requirements of outbound logistics as illustrated in Figure 4. Each observable property is observed by a specific type of sensor, this relationship is modeled in the ontology by the object property *SOSA:isOservedBy* (inverse of *SOSA:observes*).

IOT4Log includes also classes and relationships to model the behavior of actuating devices, called actuators, which perform (actuating) procedures to change the state of the world. An action is performed by an actuator and produces a result. To refer to the activity that the actuators can perform, we will use in our ontological representation model the term ***SOSA: Actuation***. *SOSA: actuatedProperty* object property links the actuation to the modified property modeled by *SOSA: ActuatableProperty* object property, and to define the actuator used to perform this actuation, we use the object property *SOSA: madeByActuator*. Therefore, the result of the actuation is the output of an actuator, and it is modeled via the object property *IOT4Log: hasResult* as shown in Figure 2.

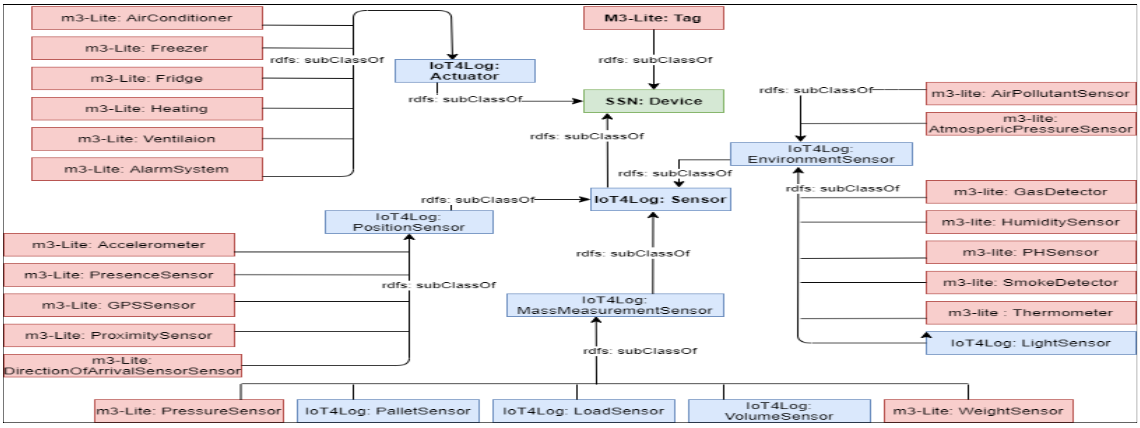


Figure 3: Device, Sensor, Actuator and related subclasses

IOT4Log: Actuator class models information about actuator which describes in our context a device making changes in the environment of logistics objects, especially products; this process is generated in response to the input resulting from sensor observations. To model this device, the class Actuator is used and it was specialized by some classes of *M3-Lite: ActuatingDevice* and by others actuators that we have added. Moreover**,** ***SOSA: ActuatableProperty*** represents the property that has been modified in response to an actuation performed by an actuator such as the state of an air conditioner

(On or off), the state of the alarm system (On or off), the state of the ventilation (open or closed), so, as shown in Figure 3 we have specialized it with new subclasses such as *AirConditionerState*, *FreezerState*, *AlarmSystemState*... etc. *SOSA: usedProcedure* object property is used to describe the relationship between the operation and the procedure used to perform this operation. The result obtained by each actuation is modeled by *SOSA:Result* class and linked to the *SOSA:Actuation* by the object property *IOT4Log:isResultOf* (inverse of *IOT4Log:hasResult*).

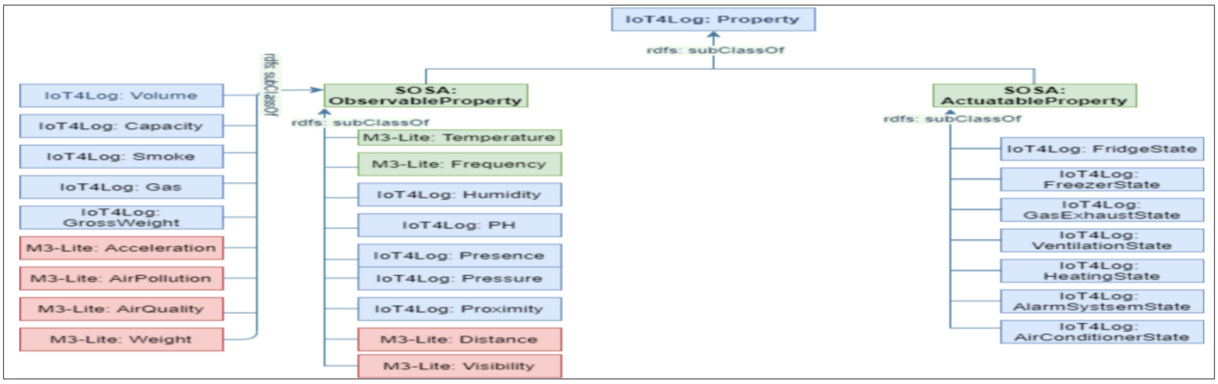


Figure 4: Class hierarchy of property concept

The *SOSA:hasSimpleResult* data type property is used to describe the simple case that only requires a literal value. The *M3-Lite: Unit* class is reused from the M3-Lite ontology to indicate the unit of measurement for the value of the result of each actuation; it is linked to the *SOSA:Result* class using the *IOT4Log:hasUnit* object property.

As each sensor and actuator concerns respectively an observable property and an actuatable property, we have defined the *IOT4Log: Property* as a superclass of the two properties reused from SOSA ontology: *SOSA: ObservableProperty* and *SOSA: ActuatableProperty* as described in the Figure 3. The object and data properties used for actuation operation are described also in Table 1.

Table 1: Objects and data properties used in IOT4Log

Relation	**Type of Property**	**Domain**	**Range**	**Inverse**	**Description**
SOSA : isObservedBy	Object Property	Observable-Property	Sensor	*observes*	Relation linking an ObservableProperty to the specific sensor used
SOSA: madeBySensor	Object Property	Observation	Sensor	*madeObservation*	Relation linking an Observation to sensor used in this operation
IOT4Log: hasResult	Object Property	Observation Actuation	Result	*isResultOf*	Relation linking the Observation and the Actuation to their result
IOT4Log: hasUnit	Object Property	Result	Unit	--	Relation linking the Result to its unit
M3-Lite: hasMeasurementType	Object Property	Device	Measurement-Type	--	Relation linking the Device to the MeasurementType used in the observation or actuation
SOSA: resultTime	Data Property	Observation Actuation	xsd: dateTime	--	Indicates the time in which observation or actuation were occurred
SOSA: hasSimpleResult	Data Property	Observation Actuation	xsd: string	--	Indicates simple result of an actuation or observation
IOT4Log: actuatedProperty	Object Property	Actuation	Actuatable-Property	--	Relation linking the actuation to the property concerned
SOSA: usedProcedure	Object Property	Sensor Actuator	Procedure	--	Relation linking the sensor or actuator to the procedure used by this device
SOSA: madeByActuator	Object Property	Actuation	Actuator	*madeActuation*	Relation linking an actuation to the specific actuator used
IOT4Log: actsOnProperty	Object Property	Actuator	Actuatable-Property	*isActedOnBy*	Relation linking an actuator to its ActuatableProperty
SOSA: observedProperty	Object property	Observation	ObservableProperty	--	Relation linking the observation to the property concerned

4.3.2. OLP-Onto ontology

The objective of the OLP-Onto ontology is to provide different concepts considered relevant to describe the physical flow containing different logistic objects ((products, logistic equipment, vehicles, transporters...) and different steps necessary to ensure the whole outbound logistics processes. According to ontology requirements specification and after having analyzed the ontologies selected for reusing, we can argue that the outbound logistics ontology could be defined with two main modules: “the logistic resource” and “the process and state”. The class hierarchy of outbound logistics ontology and the main entities and relations which are involved are presented in Figure 10 and described in the following sections.

4.3.2.1. Logistic resource module

OLP-Onto: LogisticObject class has been defined to model the concepts which represent physical resources involved in the different outbound logistics activities. It captures the concepts requisite to describe in detail the type of goods stored or transported, the equipment used for packaging, loading and all resources needed for their movement and transportation. *OLP-Onto: LogisticObject* is specialized with five components: *TradeItem*, *PackagingUnit*, *LogisticUnitLoad*, *TransportEquipment*, *TransportMeans*. The following text and Figure 5 describe the subclasses of *OLP-Onto: LogisticObject* and the associated concepts.

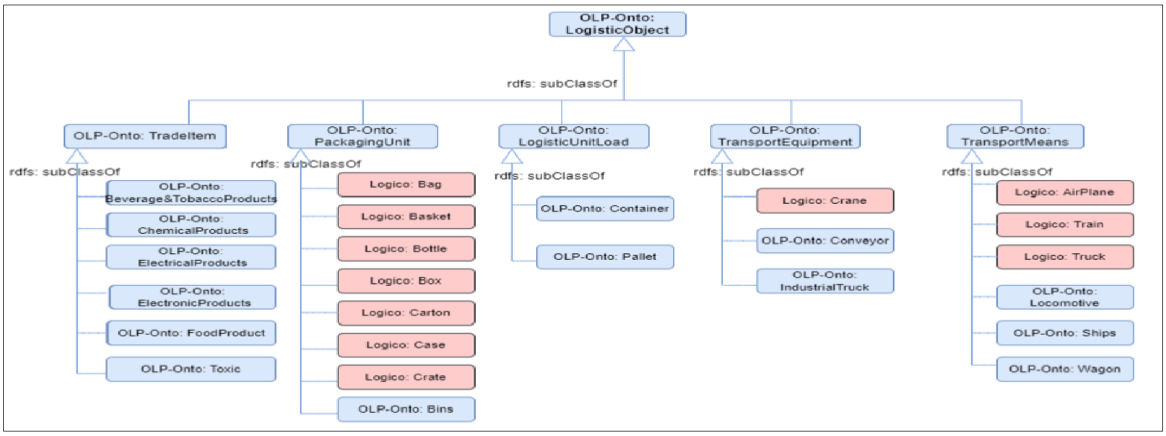


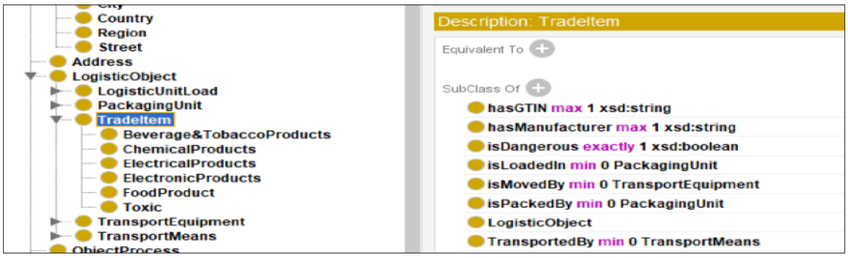
Figure 5: Logistic Object subclasses and associated concepts

Table 2: Object properties, domain, range, and the definition of main object properties used between Logistic object subClasses

Relation	Type of property	Definition	Domain	Range
isPackedBy	Object Property	Relation linking a product to unit in which it was packed	TradeItem PackagingUnit	PackagingUnit
isLoadedIn	Object Property	Relation describing logistic unit load using to load product, packaging unit or a logistic unit load as loading pallet in a container	TradeItem, PackagingUnit, LogisticUnitLoad	LogisticUnitLoad
isMovedBy	Object Property	Relation linking some logistic objects(product, packaging unit, logistic unit load) to equipment of transport used to move it from one location to another	TradeItem, PackagingUnit , LogisticUnitLoad	TransportEquipment
TransportedBy	Object Property	Relation between logistic objects(product, packaging unit, logistic unit) and means of transport used for their transport	TradeItem, PackagingUnit, LogisticUnitLoad	TransportMeans
isIn	Object Property	Relation indicating the symbolic location of objects used in outbound logistics processes	LogisticObject	SymbolicLocation
hasGTIN	Data Property	A property which identify each trade item by a Global Trade Item Number	TradeItem	xsd: string
hasName	Data Property	A property that assign a name for each logistic object	LogisticObject	rdfs: Literal
hasManufacturer	Data Property	A data property which describes the manufacturer of logistic object	LogisticObject	rdfs: Literal
hasCodeIso	Data Property	A property which define an ISO code to logistic object	LogisticObject	rdfs: Literal
hasDescription	Data Property	A property which describe each logistic object	LogisticObject	rdfs: Literal
hasGRAI	Data Property	A data property that identify logistic unit load by a Global Returnable Asset Identifier	LogisticUnitLoad	xsd: string

In *OLP-Onto*, the *OLP-Onto*: *TradeItem* class is created as a specialization of *OLP-Onto: LogisticObject* to describe information about goods involved in the concerned SC; it includes the main product types that can be used. Each trade item is identified by a Global Trade Item Number (GTIN) modeled by *OLP-Onto:hasGTIN* data property. The *OLP-*

Onto: hasMPN data property was also used to identify the trade item from the perspective of a particular manufacturer. We have defined the product type based on the Standard Classification Goods (SCG)* that aims to classify products according to their physical characteristics. On the other hand, our contribution is especially dedicated to sensitive goods that have specific features to be taken into consideration during storage and transportation processes. As a result, six specializations of *OLP-Onto*: *TradeItem* have been defined as shown in Figure 5. In addition to the properties which identify each product, the item can be traced and tracked using the packaging units, loading units, transport equipment and means of transport used during the product life cycle in the concerned SC. Object properties are used to ensure this goal such as *isPackedBy*, *isLoadedIn*, *isMovedBy*, *transportedBy*. *IsPackedBy* defines the transitive object property to indicate the unit used for packing the product, *isLoadedIn* determines the unit used for loading the product. The other properties are defined in Table 2. Figure 6 shows a description of the *TradeItem* class.



hasGTIN max 1 xsd:string
hasManufacturer max 1 xsd:string
isDangerous exactly 1 xsd:boolean
isLoadedIn min 0 PackagingUnit
isMovedBy min 0 TransportEquipment
isPackedBy min 0 PackagingUnit
LogisticObject
TransportedBy min 0 TransportMeans

Figure 6: Description of the TradeItem class

OLP-Onto: PackagingUnit groups types of objects used for the packaging of products. Each packaging unit is modeled by an ensemble of data properties as described in Table 2. The majority of Packaging unit types defined are reused from LogiCo ontology.

OLP-Onto: LogisticUnitLoad describes the materials used to combine and load items into single "units" during different logistics processes, and they can then be moved by transport equipment. Each logistic unit load is identified by a Global Returnable Asset Identifier which is modeled by the data property *OLP-Onto: hasGRAI*. The major types of these materials are pallets and containers. In Figure 7 are shown major types of containers.

OLP-Onto: TransportEquipment encompasses concepts which represent materials used to move some types of logistic objects from one place to another for their handling; they are three main types: industrial trucks, conveyors and cranes which differ from each other on how they can move within areas and paths. In Figure 7 are shown major types of conveyors and industrial trucks.

OLP-Onto: TransportMeans presents the vehicles used to transport the products and the related logistic objects. Some of *TransportMeans* subclasses are reused from LogiCo ontology.

The *LogiCo: PhysicalAttribute* class described in the LogiCo ontology includes a variety of physical properties of logistic resources, but they only measure the initial properties of the objects without taking into account the changes that can occur for different products and their environment. Thus, we modified this concept in our contribution into two entities:

OLP-Onto: PhysicalAttribute includes the main concepts used to describe the initial physical properties and dimensional information of logistic objects such as height, length, width, size, volume, etc. These attributes will be defined using object properties (*hasHeight*, *hasSize*, *hasVolume*...etc.), and data properties (*hasHightValue*, *hasSizeValue*, *hasVolumeValue*...etc.).

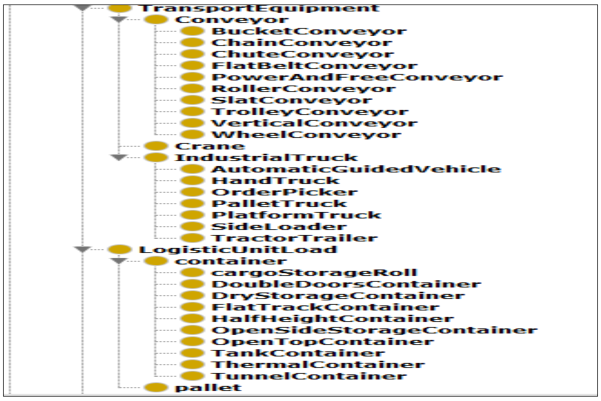


Figure 7: Types of transport equipment and logistic unit load

---

* https://www.statcan.gc.ca/eng/subjects/standard/scg/scgindex

OLP-Onto: Requirements includes the requirements needed during the logistic process; it concerns the environmental conditions that must be respected in the storage and transportation process as humidity maximal, temperature maximal. It also includes requirements related to the equipment used for loading, storage and transportation as the maximum weight of a pallet. These requirements will be used to define actions; for example, if the temperature sensor detects that the product's environment (warehouse, container, vehicle...) exceeds the maximum temperature determined in the products requirements, an alarm detecting abnormal situations in the product state would be triggered. These requirements are defined using object properties (*hasMaxHumidity*, *hasMaxTemperature*, *hasMinTemperature*,...etc.) and data properties (*hasMaxHumidityValue*, *hasMaxTemperatureValue*, *hasMinTemperatureValue*,...etc.).

OLP-Onto: Symboliclocation represents the facilities used during logistic processes to indicate the symbolic location of logistic objects as illustrated in Figure 8. The *Symboliclocation* class is connected to the *OLP-Onto: Address* class, which in turn is connected to the *OLP-Onto: GeoLocation* to specify the address of these facilities (Country, Region, City, Street), using the object properties *hasAddress* and *hasGeoLocation*, in addition to the other object and data properties used in this context (*hasCity*, *hasCountry*, *hasCountryName*, *hasStreetName*, *hasStreetNumber*, *hasPostalCode*...).

Table 3: Object properties linking logistics objects with requirements, physical parameters and address

Object property	Definition	Domain	Range
hasRequirements	Relation linking logistic object to its special requirements defined in function of its properties	LogisticObject	Requirements
hasPhysicalAttribute	Relation between logistic object and physical attribute which describe its properties	LogisticObject	PhysicalAttribute
hasGeoLocation	Linking the address to its geographic location (Country, city, street, region)	Address	GeoLocation
hasAddress	Relation linking a static resource to its address	SymbolicLocation	Address

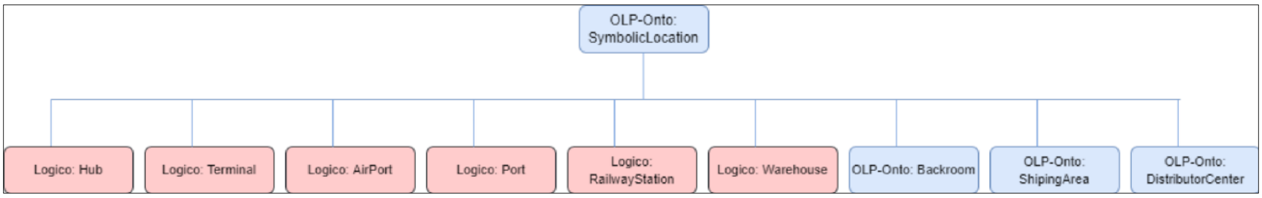


Figure 8: subclasses of SymbolicLocation

4.3.2.2. The process-state module

OLP-Onto:Process, this class represents the different types of structured sets of activities and relevant operations that have to be performed in order to process the products from their reception as semi-finished products to their delivery to the final customers.

OLP-Onto:State, this class determines different situations that logistics objects can take during outbound logistics processes. These two concepts were specialized based on the categorization of logistics processes and events in the model proposed by (Lian et al., 2007), in addition to some subclasses of the *LogiServ:Activity*[*] class. We defined them with a set of predetermined possibilities, which will be detailed in the following sections, defining when and where the process and state took place and which object is involved. Figure 9 shows the taxonomy of processes and states used in the OLP-Onto ontology.

---

[*] https://ontology.tno.nl/logiserv/

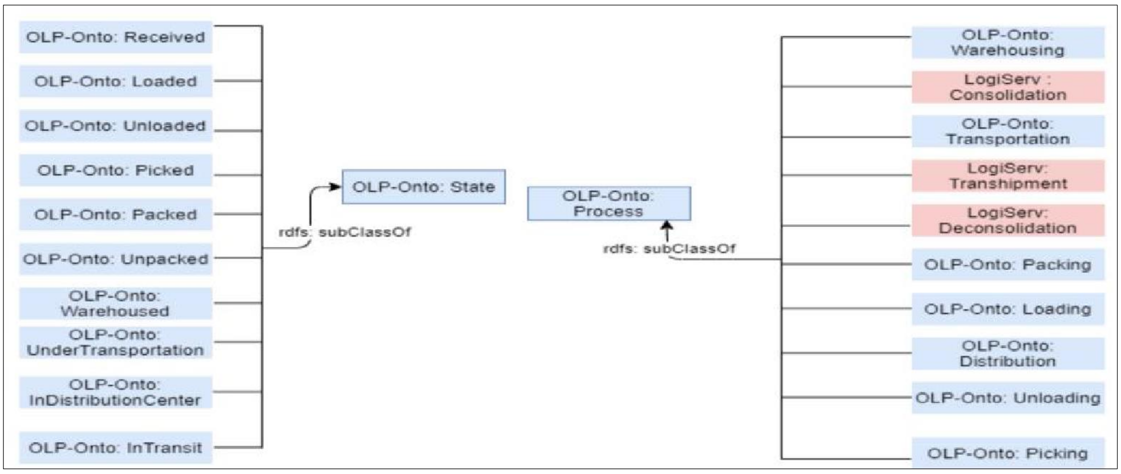


Figure 9: Taxonomy of Processes and States used in OLP-Onto

4.3.2.3. Integration of modules

As any logistic process implies a physical flow composed of a set of products, and various equipment that ensure the objective of this process, *OLP-Onto: ObjectState* and *OLP-Onto: ObjectProcess* two new classes have been added in our ontology in order to define which logistic object is concerned by a state or a process, in addition to specifying when and where it has occurred as we will see in the OLP-IOT ontology. These two classes determine the different states and processes that the logistic object can. They are linked to the two modules Resource and Process-State by using three object properties: *OLP-Onto: isInState*, *OLP-Onto: isInProcess* and *OLP-Onto: concerns*. *OLP-Onto: isInState* determines the state that a logistic object can take in a given moment, while linking this instance of the *OLP-Onto: ObjectState* class to the concerned logistic object via the *OLP-Onto: concerns* object property. In the same way *OLP-Onto: isInProcess* captures the process to which the logistics object belongs (this object is linked to the instance of the *OLP-Onto: ObjectState* class via the *OLP-Onto: concerns* object property). The definition of the object properties used is shown in Table 5.

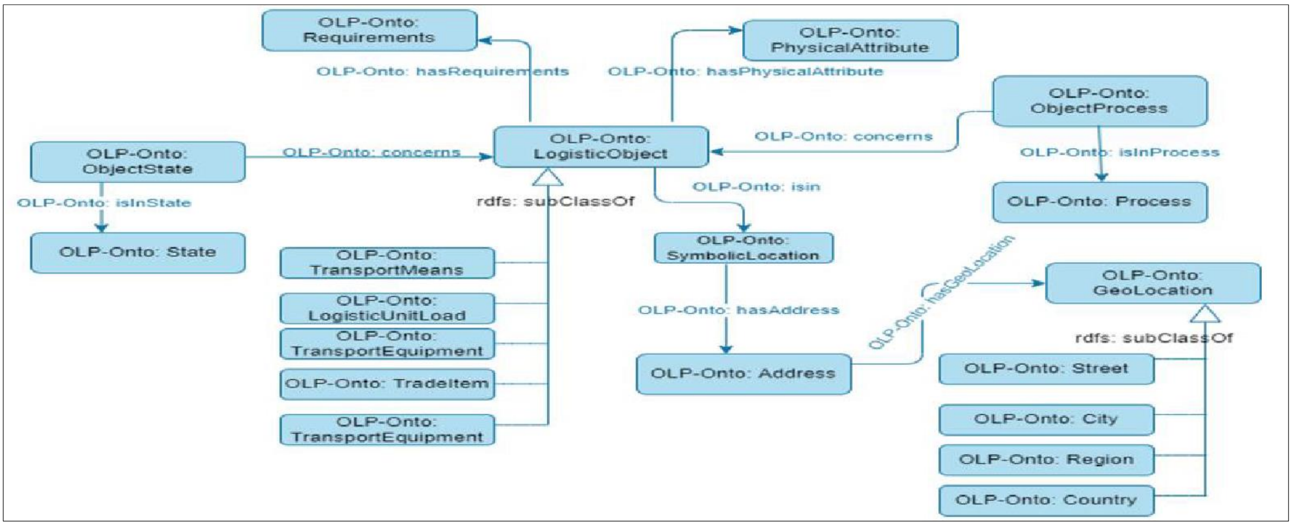


Figure 10: Global architecture of OLP-Onto

Table 4: Object properties used to link the two modules of OLP-Onto

Object property	Definition	Domain	Range
isInState	A relation to define which state is concerned by an instance of objectState class	ObjectState	State
isInProcess	A relation to define which process is concerned by an instance of objectProcess class	ObjectProcess	Process
concerns	A relationship to define which logistic object is concerned by a specific state or process.	ObjectState ObjectProcess	LogisticObject

4.3.3. OLP-IOT ontology

In this section we will present the global ontology OLP-IOT developed following the integration and fusion of the two modules IOT4Log and OLP-Onto. This ontology allows tracking and tracing of assets and objects during logistic processes by defining the state of the product and its physical attribute at any time. OLP-IOT reuses the IOT4Log and OLP-Onto ontologies that we have already described in the previous sections and a **Spatio-temporal** module that we will detail in the next section. The global modular ontology is enriched by inter-module relations in order to integrate them semantically to have an ontological model allowing the modeling of the context in the domain of logistics based on IOT. Figure 14 gives an overview of OLP-IOT revealing the different modules and presenting some inter and intra-module relations.

4.3.3.1. The Space-Time module

In IOT-based logistics systems, time is a key factor in tracking and monitoring logistics processes, and therefore ensuring the reliability and quality of stored and transported products. The stakeholders must be able to interpret the sensor observations in specific time frames, in order to react and respond at the right time to abnormal events that may occur. In this context, the temporal aspects (when an observation was made or transferred) must be well described. To model this, we reuse the ***Time*** ontology which allows describing temporal relations and properties. It also supports durations and time intervals; these features are advantageous when describing complex event specifications and imprecise measurement times.

Location is a fundamental element in IOT-based logistics systems. This concept describes different types of location, in particular, symbolic and geographic Location. In our context, we reuse the **GeoSPARQL**[*]; an ontology providing concepts and relations that allow representation of physical spatial locations and the relations between them. Most of the concepts represented in this ontological model are entities that can be located in space, such as sensors, actuators, logistic objects and processes that can be located via their participating entities. The *geo:SpatialObject* class contains two subclasses called *geo:Feature* and *geo:Geometry*. The *geo:Feature* class represents 3D objects or 2D areas and can be assigned geometries that describe them through the *geo:hasGeometry* property.

4.3.3.2. Integration of all modules

The integration between modules is mainly done through the use of relations (object properties, equivalence) that link the concepts of a module to those of other modules as described below:

In the context of the OLP-IOT ontology, the first step to ensure the traceability of a product throughout the outbound logistics processes starts with the incorporation of each product by a unique RFID tag, this identification is essential to transform the product from a simple object to a smart object that can communicate and exchange information with other devices or systems. For outbound logistics, the products received from previous operations and their associated RFID must be registered in the OLP-IOT ontology. Since products can be packed, loaded and moved from different locations using different equipment, identification of this involved equipment is necessary to ensure high visibility of product transition during all logistic processes.

As a result, the logistic objects in the OLP-Onto ontology (OLP-Onto: LogisticObject) will be linked to the RFID tag (IOT4Log: RFID) by OLP-IOT: TaggedBy relationship. This object property, created in the OLP-IOT ontology to constitute the first integration of the IOT in outbound logistics. The object properties defined in the OLP-Onto ontology such as *OLP-Onto: isPackedBy*, *OLP-Onto: isLoadedIn*, *OLP-Onto: isMovedBy*, *OLP-Onto: isTransportedBy* provide product traceability; as they identify respectively: in which packaging unit the product was packed, which logistics unit was used to load the product, which equipment was used to move the product from one place to another and in which means of transportation it was transported. This traceability is ensured by using these relations as well as inference rules to deduce new knowledge, as we will see in the inference part. Moreover, the association of logistic objects with an RFID tag ensures the identification of all processes involved and all states the product can take by using the object properties *OLP-Onto: isInState* and *OLP-Onto: isInProcess*.

A crucial step to ensure product traceability and safety is the deployment of sensors and actuators in different objects and logistic locations. In this context, the two object properties: *OLP-IOT: isHostedOn* and *SOSA: hasFeatureOfInterest* are used to link the two ontologies.

OLP-IOT: IsHostedOn models the relationship between the classes that represent the IOT devices (*IOT4Log: Sensor*, *IOT4Log: Actuator*) and the location in which they are hosted. They can be present in an *OLP-Onto: SymbolicLocation* (e.g. a temperature sensor exists in a warehouse to detect the temperature of this location) or included in a special *OLP-Onto: LogisticObject* (e.g. a sensor contained in an *AutomaticGuidedVehicle* a subclass of *OLP-Onto: IndustrialTruck* to detect the presence of a pallet)

The *OLP-Onto: LogisticObject* and *OLP-Onto: SymbolicLocation* classes are linked to the Location module by the *OLP-IOT: hasLocation* relation. This relation links an instance of the concerned logistic object or a symbolic location with the

---

[*] http://www.opengis.net/ont/geosparql

class *geo:SpatialObject*. This relation indicates the exact location of a product, logistic equipment or even a symbolic location like warehouses and distribution centers...etc.

The *OLP-Onto: ObjectState* class is linked to the Location module by the *OLP-IOT: hasLocation* relation. This relation links an instance of the object state with the *geo:SpatialObject* class. This relation is used to specify the location in which the object state occurs. Thus, it should be noted that this location can be deduced using the location in which the observation of the object concerned was produced.

The two classes *SOSA: Observation* and *SOSA: Actuation* are linked to *OLP-Onto: LogisticObject* by the relation *SOSA: hasFeatureOfInterest*. It has been reused to link the detection and reaction operations to the objects involved in these operations, thus, it provides a semantic representation of the environmental condition in which the product and its environment are contained. As we cited in the IOT4Log ontology, the *SOSA: Observation* class is related to *SOSA: ObservableProperty*, therefore, a semantic tracing and tracking of the involved products and objects will be ensured by recording the results collected by the sensors on the physical properties of the object (such as temperature of a product, humidity in a vehicle, presence of a gas in a warehouse...). As we have presented in the OLP-Onto ontology, *LogisticObject* can have particular requirements to make sure that it goes through all the involved processes in good conditions. The relationship established between the object and device-observation ensures that the requirements of the logistic objects will be met and tested. This is done by comparing the initial requirements of each logistic object with the value of its physical properties that are recorded with sensors along the outgoing logistic processes. Figure 11 illustrates the integration of IOT4Log with the OLP-Onto ontology using the relationships: *OLP-IOT: IsHostedOn*, *SOSA: hasFeatureOfInterest* and *OLP-IOT: TaggedBy*.

In the OLP-IOT ontology, the IOT4Log and OLP-Onto ontologies are linked to the Location and Time modules as illustrated in Figure 12 and described below:

The *OLP-Onto: ObjectProcess* is linked to the Time module by the *Time: hasDuration* relation. This relation links an instance of an object process with the *Time: Duration* class. It is used to assert the duration of a process.

The *OLP-Onto: ObjectState* is linked to the Time module by the *Time: hasTime* relation. This relation links an instance of an object process with the *Time: temporalEntity* class. It is used to determine when an observation has been made.

The IOT4Log: Observation class is linked to the Time module and the Location module, using respectively the *OLP-IOT: hasLocation* and *Time: hasTime* relations. The *OLP-IOT: hasLocation* relation allows linking an observation instance (*IOT4Log: Observation*) to the location where it has been performed (*geo:SpatialObject*). For the relation *Time: hasTime*, it allows indicating the instant (*Time: temporalEntity*) in which an instance of an observation (*IOT4Log: Observation*) has been performed.

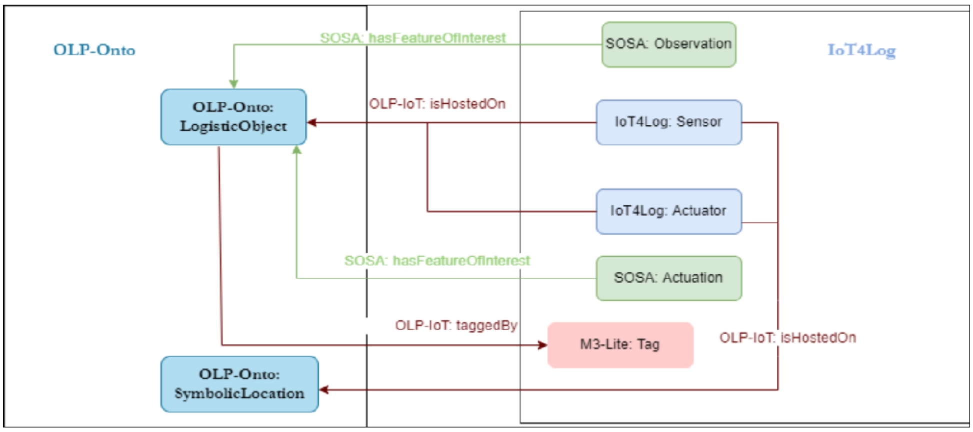


Figure 11: Object properties for the integration of IOT4Log and OLP-Onto

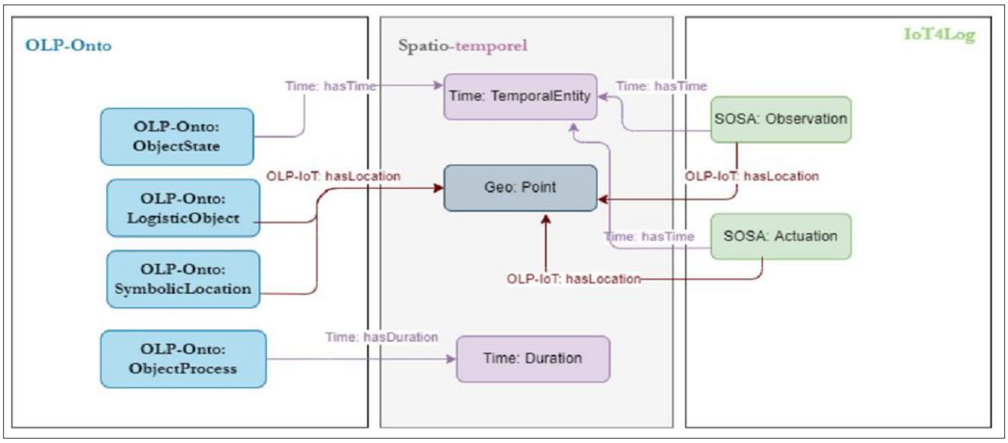
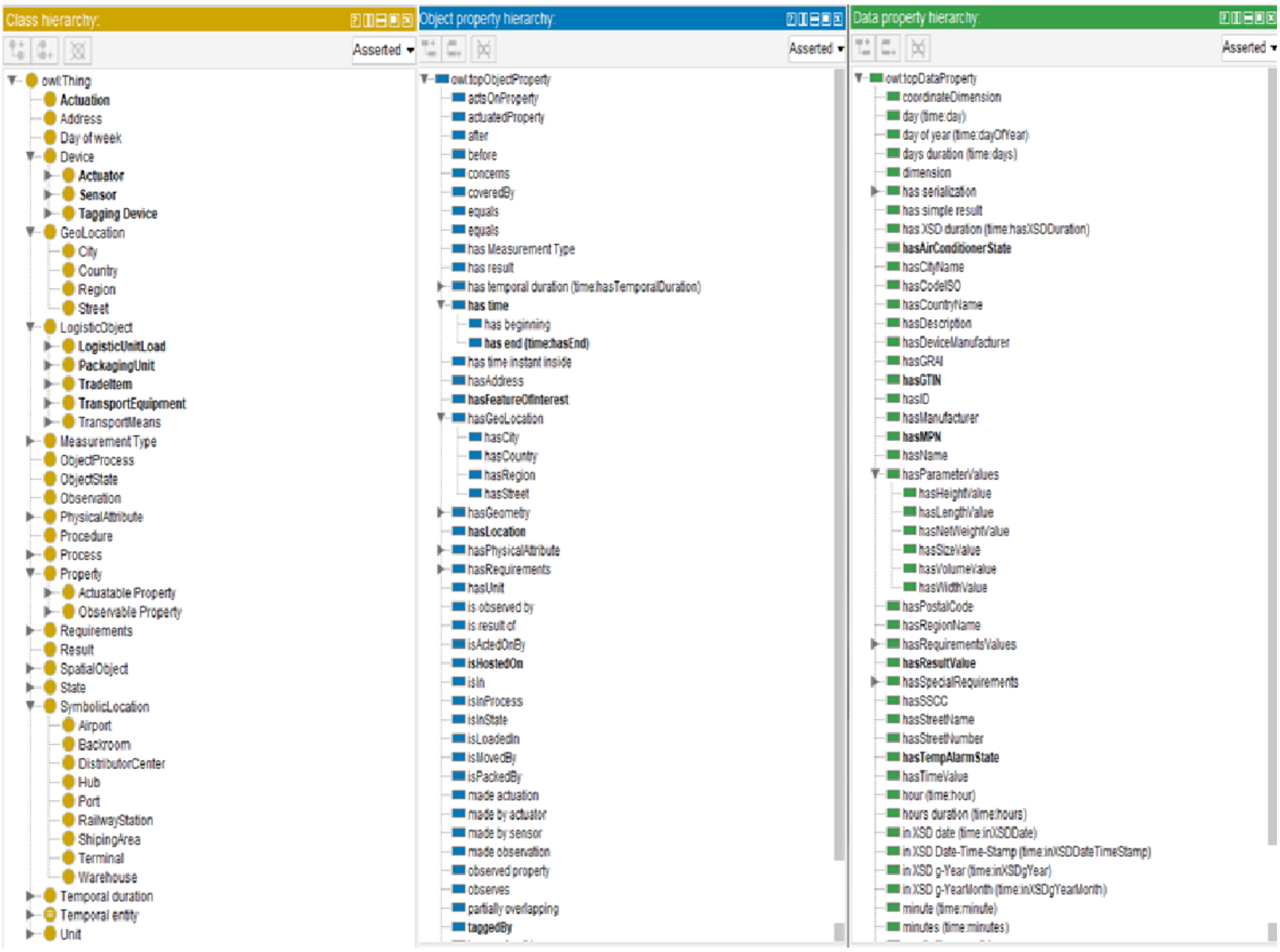


Figure 12: Integration of the space-time module with IOT4Log and OLP-Onto

The *IOT4Log: Actuation* is linked to the Time and the Location modules, using the *OLP-IOT: hasLocation* and *Time: hasTime* relations respectively. The *OLP-IOT: hasLocation* relation allows linking an instance of an actuation (*IOT4Log: Actuation*) to the location in which it has been performed (*geo:SpatialObject*). For the relation *Time: hasTime*, it allows indicating the instant (*Time: temporalEntity*) in which an instance of an action (*IOT4Log: Actuation*) was made.
Figure 14 shows an overview of the main classes and properties of the OLP-IOT ontology. Similarly, Figure 13 shows the classes, object properties and data properties of the OLP-IOT class in Protégé.



- owl:Thing
  - **Actuation**
  - Address
  - Day of week
  - Device
    - **Actuator**
    - **Sensor**
    - **Tagging Device**
  - GeoLocation
    - City
    - Country
    - Region
    - Street
  - LogisticObject
    - **LogisticUnitLoad**
    - **PackagingUnit**
    - **TradeItem**
    - **TransportEquipment**
    - TransportMeans
  - Measurement Type
  - ObjectProcess
  - ObjectState
  - Observation
  - PhysicalAttribute
  - Procedure
  - Process
  - Property
    - Actuatable Property
    - Observable Property
  - Requirements
  - Result
  - SpatialObject
  - State
  - SymbolicLocation
    - Airport
    - Backroom
    - DistributorCenter
    - Hub
    - Port
    - RailwayStation
    - ShipingArea
    - Terminal
    - Warehouse
  - Temporal duration
  - Temporal entity
  - Unit

- owl:topObjectProperty
  - actsOnProperty
  - actuatedProperty
  - after
  - before
  - concerns
  - coveredBy
  - equals
  - equals
  - has Measurement Type
  - has result
  - has temporal duration (time:hasTemporalDuration)
  - **has time**
    - has beginning
    - **has end (time:hasEnd)**
  - has time instant inside
  - hasAddress
  - **hasFeatureOfInterest**
  - hasGeoLocation
    - hasCity
    - hasCountry
    - hasRegion
    - hasStreet
  - hasGeometry
  - **hasLocation**
  - hasPhysicalAttribute
  - hasRequirements
  - hasUnit
  - is observed by
  - is result of
  - isActedOnBy
  - **isHostedOn**
  - isIn
  - isInProcess
  - isInState
  - isLoadedIn
  - isMovedBy
  - isPackedBy
  - made actuation
  - made by actuator
  - made by sensor
  - made observation
  - observed property
  - observes
  - partially overlapping
  - **taggedBy**

- owl:topDataProperty
  - coordinateDimension
  - day (time:day)
  - day of year (time:dayOfYear)
  - days duration (time:days)
  - dimension
  - has serialization
  - has simple result
  - has XSD duration (time:hasXSDDuration)
  - **hasAirConditionerState**
  - hasCityName
  - hasCodeISO
  - hasCountryName
  - hasDescription
  - hasDeviceManufacturer
  - hasGRAI
  - **hasGTIN**
  - hasID
  - hasManufacturer
  - **hasMPN**
  - hasName
  - hasParameterValues
    - hasHeightValue
    - hasLengthValue
    - hasNetWeightValue
    - hasSizeValue
    - hasVolumeValue
    - hasWidthValue
  - hasPostalCode
  - hasRegionName
  - hasRequirementsValues
  - **hasResultValue**
  - hasSpecialRequirements
  - hasSSCC
  - hasStreetName
  - hasStreetNumber
  - **hasTempAlarmState**
  - hasTimeValue
  - hour (time:hour)
  - hours duration (time:hours)
  - in XSD date (time:inXSDDate)
  - in XSD Date-Time-Stamp (time:inXSDDateTimeStamp)
  - in XSD g-Year (time:inXSDgYear)
  - in XSD g-YearMonth (time:inXSDgYearMonth)
  - minute (time:minute)
  - minutes (time:minutes)

Figure 13: OLP-IOT classes and properties hierarchy

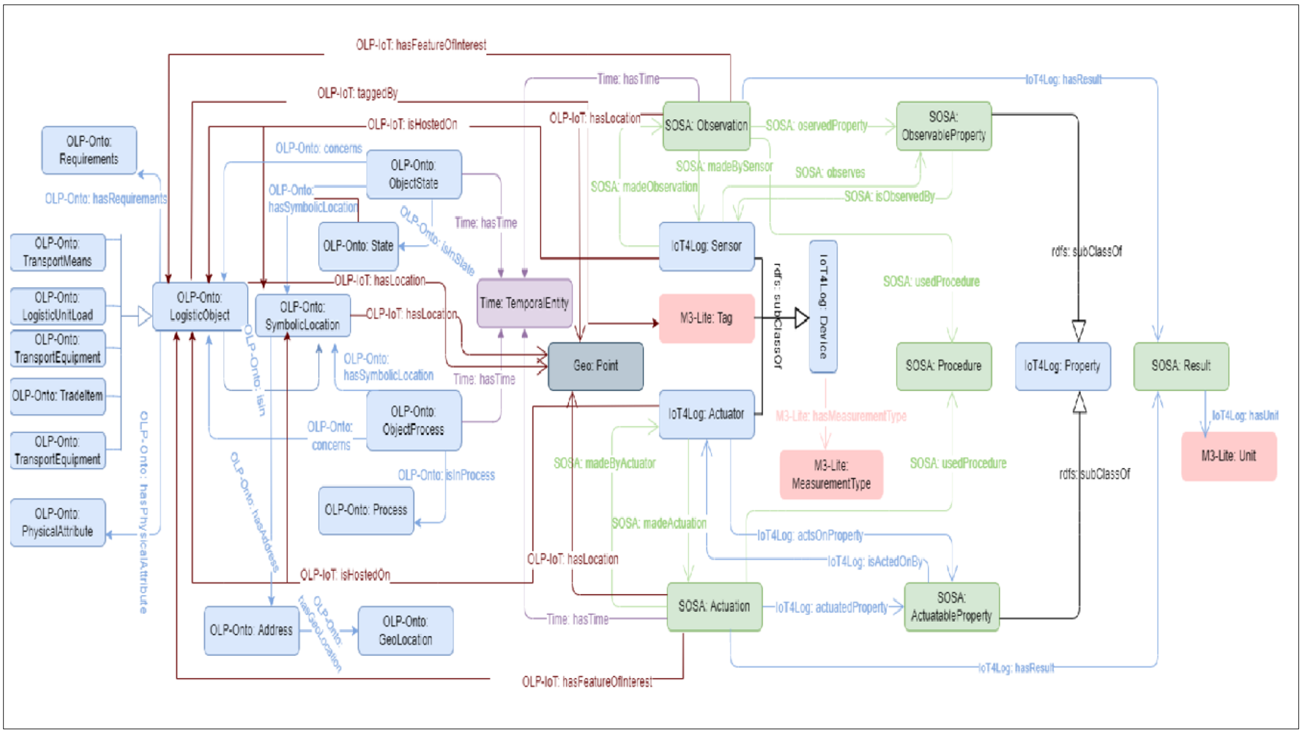


Figure 14: Global architecture of the OLP-IOT

5. Inferences and evaluation

Ontology evaluation is an important task that is necessary to ensure that what is built meets the application requirements and allows users to appreciate the quality of the ontologies. In this context, and after developing the OLP-IOT ontology, an evaluation process was performed to validate and verify the quality of the resulting ontology. In our research, we present the evaluation of OLP-IOT through the verification step using metrics followed by the validation step using criteria.

5.1. Verification

Ontology verification consists of ensuring that the ontology is constructed correctly and it is consistent. It answers the question, "Are we producing the ontology correctly?" ? (Suárez-Figueroa et al., 2012). The goal is to ensure also that the classes are satisfactory, and that the inferred model reflects the intended semantics. To do this, we checked a Three-step process. First, we used the Pellet, which is an OWL reasoner built into the Protégé software, to determine the consistency of the ontological approach developed. By using Pellet; we ensured that the ontology no longer has logical inconsistencies. Second, to check whether the ontology meets the already determined specifications, we compare the developed ontological approach with the specifications already provided in section 4.1. To do this, we translated the competency questions (CQs) into SPARQL[*] language to query the ontology. In this paper we present the example of CQ4. To evaluate this issue, we exploit a real-world dataset from Walmart; an American multinational retail company that operates a chain of hypermarkets (also known as supercenters). We instantiate our ontology with data from some dairy products with refrigeration conditions as shown in Table 5. We assume that Walmart in our case represents a distribution center containing dairy products. The objective is to display the products, their GTINs, and their maximum and minimum temperature requirements. The result of this query is shown in Figure 15.

The OntoMetric (Lantow, 2016) tool has been chosen to evaluate the quantitative quality metrics of OLP-IOT ontology as it offers a free online environment to upload the OWL file, calculate ontology metrics and generate an XML file of calculated ontology metrics. In our context, OLP-IOT was evaluated based on three main metrics: Base metric, Schema metric, and Graph metric. This evaluation includes also calculating metrics of IOT4Log and OLP-Onto ontologies as they are modules of OLP-IOT ontology. The base metrics of OLP-IOT ontology and its modules are listed in Table 6**,** for Schema and Graph metrics compared with other ontologies as SOSA and LogiCo are presented in Table 7. As we have not found in the literature review an ontology which responds exactly to our requirements, comparison was divided into two parties: one concerned a comparison between LogiCo and OLP-Onto and another one to compare SOSA and IOT4Log. As it can be observed from Table 6, most metrics calculated affirm that the ontologies developed are very rich

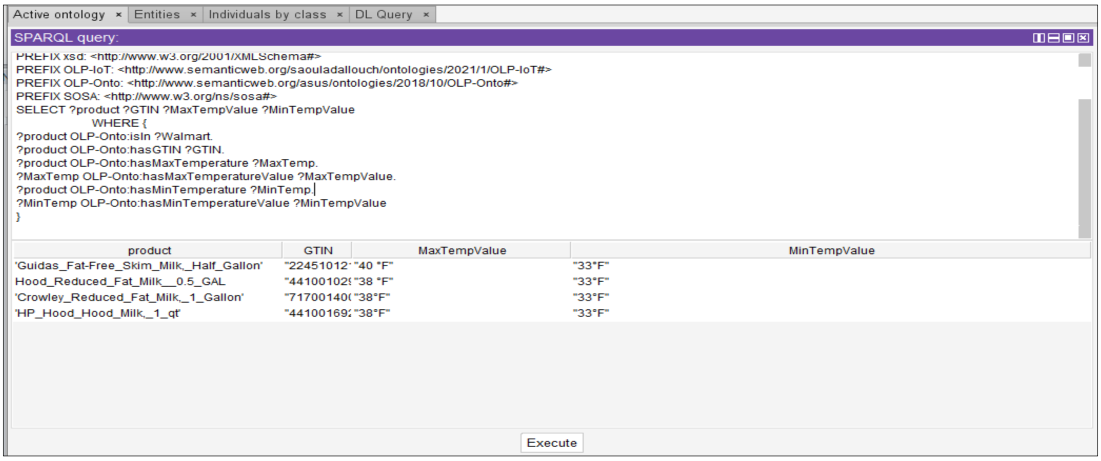
---

[*] https://www.w3.org/TR/rdf-sparql-query/

regarding number of classes and properties integrated which proves that they represent well both IOT and logistics domains.

Table 5: A data set from a Walmart retail center

Nom du produit	**Code GTIN**	**Catégorie**	**Température requise**
Guidas Fat-Free Skim Milk, Half Gallon	22451012104	Food| Fresh Food| Dairy, Eggs& Cheese| Milk& Cream	33- 40°F
Hood Reduced Fat Milk 2%, 0.5 GAL	44100102936	Food | Fresh Food | Dairy, Eggs & Cheese | Milk & Cream	33–38 °F.
Crowley 2% Reduced Fat Milk, 1 Gallon	71700140013	Food | Fresh Food | Dairy, Eggs & Cheese | Milk & Cream	33– 38 °F.
HP Hood Hood Milk, 1 qt	44100169236	Food | Fresh Food | Dairy, Eggs & Cheese | Milk & Cream	33– 38°F.



```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX OLP-IoT: <http://www.semanticweb.org/saouladallouch/ontologies/2021/1/OLP-IoT#>
PREFIX OLP-Onto: <http://www.semanticweb.org/asus/ontologies/2018/10/OLP-Onto#>
PREFIX SOSA: <http://www.w3.org/ns/sosa#>
SELECT ?product ?GTIN ?MaxTempValue ?MinTempValue
            WHERE {
?product OLP-Onto:isIn ?Walmart.
?product OLP-Onto:hasGTIN ?GTIN.
?product OLP-Onto:hasMaxTemperature ?MaxTemp.
?MaxTemp OLP-Onto:hasMaxTemperatureValue ?MaxTempValue.
?product OLP-Onto:hasMinTemperature ?MinTemp.
?MinTemp OLP-Onto:hasMinTemperatureValue ?MinTempValue
}
```

```
                 product                  GTIN      MaxTempValue        MinTempValue
'Guidas_Fat-Free_Skim_Milk,_Half_Gallon'  "22451012 "40 °F"             "33°F"
Hood_Reduced_Fat_Milk__0.5_GAL            "44100102 "38 °F"             "33°F"
'Crowley_Reduced_Fat_Milk,_1_Gallon'      "71700140 "38°F"              "33°F"
'HP_Hood_Hood_Milk,_1_qt'                 "44100169 "38°F"              "33°F"
```

Figure 15: Result of the search for products and their maximum and minimum temperature required in the "Walmart" distribution center

Relationship Richness (RR) metrics for example in IOT4Log and OLP-Onto are not close to zero which indicates that they are rich and contains relationships other that class-subclass regarding LogiCo, for SOSA, its RR value is 1, as it is general and does not contain a specialization of classes. For Inheritance Richness (AR) metrics measured, they indicate that our ontologies are vertical and reflect a detailed type of knowledge, and this is due to the fact that our ontologies describe in detail the domain of logistics and IOT.

Table 6: Base metrics of OLP-IOT and its modules

Metrics	**IOT4Log**	**OLP-Onto**	**OLP-IOT**
Axioms	1000	1355	163
Logical axioms count	331	508	109
Class count	182	120	79
Object property count	16	30	12
Data property count	5	32	3
Properties count	21	62	15
Individual count	0	0	16
			16
DL expressivity	ALCI(D)	ALCH(D)	ALCH(D)
Class axioms			
SubClassOf	173	124	51
EquivalentClasses	0	0	0
DisjointClasses	111	233	17
GCI count	0	0	0
Hidden GCI Count	0	0	0
Object property axioms			
SubObjectPropertyOf	0	15	1
Equivalent object properties	0	0	0
Inverse object properties	5	0	0
Disjoint object properties	0	0	0
Functional object properties	0	0	0
	0	0	0

Inverse functional object properties	0	0	0
Transitive object property	0	0	0
Symmetric object property	0	0	0
Asymmetric object property Reflexive	0	0	0
object property	0	0	0
Irreflexive object property	16	29	6
Object property domain	16	29	6
Object property range SubPropertyChainOf	0	0	0
Data property axioms			
SubDataPropertyOf	0	14	0
Equivalent data properties	0	0	0
Disjoint data properties	0	0	0
Functional data property	0	0	0
Data property domain	5	30	2
Data Property range	5	31	3
Individual axioms			
Class assertion axioms count:	0	0	16
Object property assertion	0	0	3
Data property assertion	0	0	2
Negative object property assertion	0	0	0
Negative data property assertion	0	0	0
Same individuals	0	0	0
Different individuals	0	0	0

Table 7: Schema and Graph metrics comparison

Metrics	**IOT4Log**	**OLP-Onto**	**OLP-IOT**	**LogiCo**	**SOSA**
Schema Metric					
Attribute richness	0.027473	0.266667	0.037975	0.304348	0.125
Inheritance richness	0.950549	1.033333	0.64557	1.413043	0.0
Relationship richness	0.423333	0.679587	0.3625	0.327586	1.0
Attribute class ratio	0.0	0.0	0.0	0.0	0.0
Equivalence ratio	0.0	0.0	0.0	0.072464	0.0
Axiom/class ratio	5.494505	11.291667	2.063291	6.144928	19.5625
Inverse relations ratio	0.3125	0.0	0.0	0.25	0.428571
Class/relation ratio	0.606667	0.310078	0.9875	0.475862	0.761905
Graph Metric					
Absolute root cardinality	9	9	41	6	16
Absolute leaf cardinality	144	97	60	95	16
Absolute sibling cardinality	182	117	78	138	16
Absolute depth	579	395	123	822	16
Average depth	3.181319	3.347458	1.5	4.697143	1.0
Maximal depth	6	5	2	7	1
Absolute breadth	182	118	82	175	16
Average breadth	4.666667	5.619048	4.315789	3.240741	16.0
Maximal breadth	32	10	41	14	16
Ratio of leaf fan-outness	0.791209	0.808333	0.759494	0.688406	1.0
Ratio of sibling fan-outness	1.0	0.975	0.987342	1.0	1.0
Tangledness	0.0	0.041667	0.037975	0.210145	0.0
Total number of paths	182	118	82	175	16
Average number of paths	30.333333	23.6	41.0	25.0	16.0

5.2. Evaluation based on the criteria

The Ontology Pitfall scanner (Poveda-Villalón et al., 2012) was chosen to evaluate the criteria-based evaluation of our developed ontologies. OOPS! is a free web-based tool used to evaluate ontology by searching and diagnosing errors in ontologies regarding a catalog of 41 common pitfalls. These pitfalls are classified into three types: minor, important and critical. In addition to detecting errors, OOPS! suggests corrections and improvements. The initial evaluation of OLP-IOT using OOPS! has asserted that no critical pitfall was detected and has generated two important pitfalls which concern untyped classes and properties; they persist due to ontology imports and reusing. Two minor pitfalls were detected as presented in Table 8. These minor pitfalls concern especially annotations and missing inverse relationships. We have decided to keep these minor errors as they do not create a problem in our ontology.

Table 8: Pitfalls detected in OLP-IOT

Pitfall N°	Description	Importance level	Cases
P08	Missing annotations	Minor	15
P13	Inverse relationships not explicitly declared	Minor	6
P34	Untyped class	Important	71
P35	Untyped property	Important	3

5.3. Inference

Reasoning is one of the main features of ontologies, as it allows inferring new information and facts based on existing knowledge and established rules. Therefore, we used our proposed ontologies and the Semantic Web Rule Language (SWRL) to establish the rules required to provide high visibility of products and their status and to increase accuracy in the outbound logistics domain. In our work, processing the data and information provided by various sensors using reasoning techniques is a key factor in ensuring the accuracy and safety of the products and objects involved. Additionally, it allows predicting events and reacting in case of abnormal situations by notifying users or triggering alarms and adequate actions. To achieve this goal, a set of SWRL rules was created based on the ontologies that we have developed, the reasoning tool (Pellet) was used for inference and reasoning. Here, some example of rules created in our ontologies and their tests in Protégé using some examples:

Rule 1: *TradeItem(?T) ^isPackedBy(?T, ?P) ^ isLoadedIn(?P, ?L) -> isLoadedIn(?T, ?L)*
Rule 1 state that if a trade item T is packed by a packaging unit P and this packaging unit is loaded in a logistic unit load L, then the trade item T is loaded also in the logistic unit load L.

Rule 2: *PackagingUnit(?P) ^ isLoadedIn(?P, ?L) ^ isMovedBy(?L, ?T) -> isMovedBy(?P, ?T)*
Rule 2 states that if a packaging unit P is loaded in a logistic unit load L and L is moved by a transport equipment T, then the packaging unit P is moved also by the transport equipment T.

Rule 3. *PackagingUnit(?P) ^ isLoadedIn(?P, ?L)) ^TransportedBy(?L, ?TM) -> TransportedBy(?P, ?TM)*
This rule states that if any packaging unit p is loaded in a logistic unit load l, and l is transported by a transport means TM, then this packaging.

Table 9 shows some instances created in test 1 to evaluate the reasoning rules described above.

Table 9: Instances created in test 1 to evaluate rule 1, 2 and 3.

Instance	Class
Carton_A	Packaging unit
Product_A	Trade item
Pallet_A	Logistic unit load
Crane_A	Transport equipment
Truck_A	Transport means

In test 1, we have conceived a case study in which a product (Product_A) was packed by a packaging unit (Carton_A); this carton needs to be loaded into a logistic unit load (pallet_A) to be moved then by transport equipment (Crane_A). This equipment will in turn be transported by a transport means (Truck_A) to arrive at its final destination. To achieve this goal in our modeling, we have defined only initial object properties that describe these processes, and then we have used the three rules described above to infer new information which describes in detail each logistic object used. For example, for the trade item (Product_A), it is connected with the packaging unit (Carton_A) by means of isPackedBy object property, but based on the available information concerning relations between others logistic objects, our model is able to make the inference that this product is loaded in (pallet_A) and is moved by (Crane_A) and finally transported by (Truck_A). The same logic is used to infer new information concerning other logistic objects as we can see in yellow in Figure 16. This figure taken from Protégé describes the reasoning mechanism used in test 1.

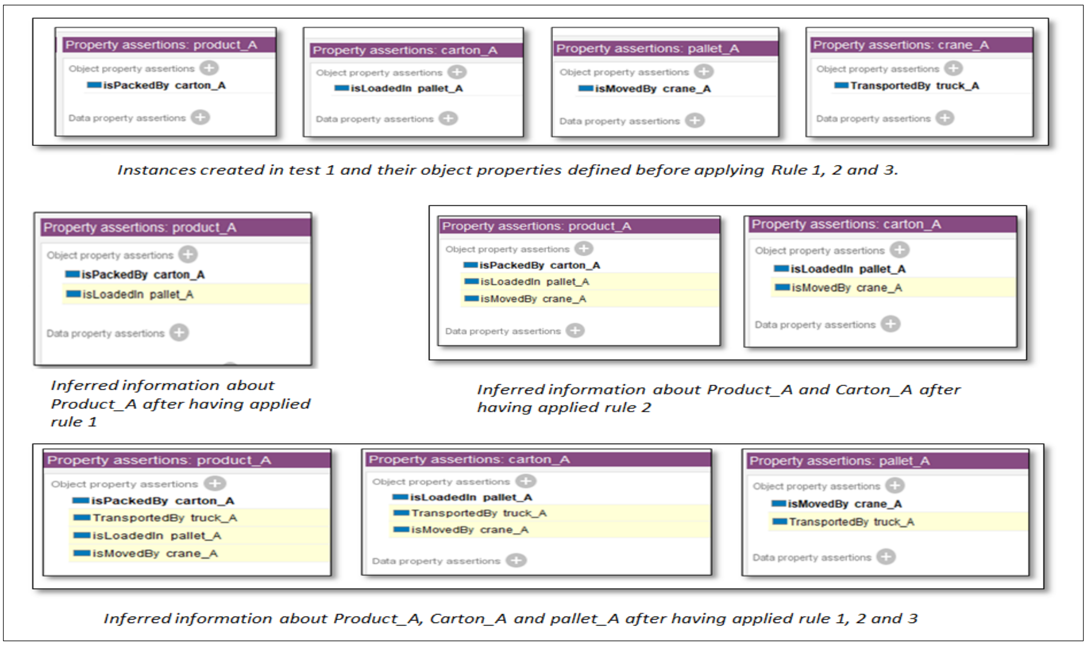


Figure 16: The reasoning mechanism used in test 1 to infer new information about logistic objects by applying Rule 1, 2 and 3.

6. Application to the case of an IOT monitoring system for the transportation of dairy product

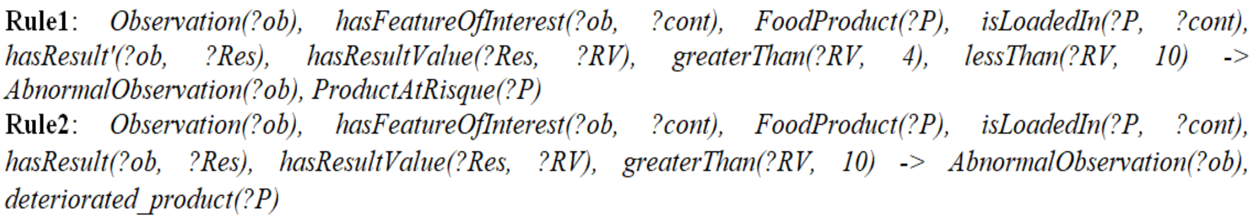
To evaluate the relevance of the presented ontological approach, we consider a use case concerning an IOT monitoring system for the transportation of dairy products, specifically cheese. Cheeses need to be protected from heat and cold. Most cheeses tend to "degrease" at 20°C (68°F) and above. During defatting, fat escapes from the paste and the cheese quickly becomes rancid. On the other hand, subjecting certain types of cheese to freezing temperatures will result in texture changes that may not be acceptable to consumers. As a general rule, cheese should not be stored at temperatures below 30°F (-1°C) or above 50°F (10°C) (Brecht et al., 2019). Recommended transport conditions for cheese are illustrated in Table 10. The table will illustrate the temperature values for three cases: transportation with better quality, transportation with conditions that may cause risk, and finally conditions that cause deterioration of this type of product.

Table 10: Recommended conditions for the transport of cheese

Situations	**Temperature**
Best conditions	1<=Temp<=4 (°C)
Risky	-1<=Temp<=1 (°C) 5<=Temp<=10 (°C)
Deteriorated product	Temp<-1 or Temp>10

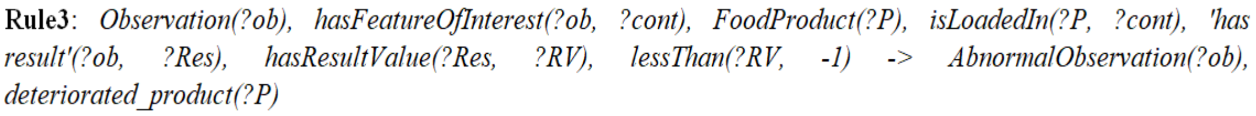
We assume that the temperature sensors are deployed in the refrigerated containers used to transport cheese. Also, the products, their packaging and the containers used for storage and transport must be equipped with RFID tags. Identification using RFID in combination with temperature detection using sensors ensures monitoring of temperature changes in the storage containers at different times. In fact, the sensors continuously transmit the temperature values measured in each container using the RFID tags to identify and locate them. In this context, ontology-based reasoning, through its rule-based extensions, allows to transform the raw observations collected by the sensors used into higher-level abstractions, such as situations of interest, which are meaningful for the actors of the so-called cheese SC. These situations can include the detection of complex events; it is essentially used to determine if the obtained measurements exceed their threshold. For example, those responsible for transporting cheese to the consignees are interested in knowing the condition of the transported products. In other words, they will be interested in knowing if there are products that are at risk of spoilage, or products that have already deteriorated due to abnormal temperature.

To handle these events, various SWRL rules are proposed and implemented. Then, the SWRL rule engine is executed by pulling the registered SWRL monitoring rules. These rules include the following:



Rule1: *Observation(?ob), hasFeatureOfInterest(?ob, ?cont), FoodProduct(?P), isLoadedIn(?P, ?cont), hasResult'(?ob, ?Res), hasResultValue(?Res, ?RV), greaterThan(?RV, 4), lessThan(?RV, 10) -> AbnormalObservation(?ob), ProductAtRisque(?P)*

Rule2: *Observation(?ob), hasFeatureOfInterest(?ob, ?cont), FoodProduct(?P), isLoadedIn(?P, ?cont), hasResult(?ob, ?Res), hasResultValue(?Res, ?RV), greaterThan(?RV, 10) -> AbnormalObservation(?ob), deteriorated_product(?P)*



Rule3: *Observation(?ob), hasFeatureOfInterest(?ob, ?cont), FoodProduct(?P), isLoadedIn(?P, ?cont), 'has result'(?ob, ?Res), hasResultValue(?Res, ?RV), lessThan(?RV, -1) -> AbnormalObservation(?ob), deteriorated_product(?P)*

Table 11: Example of instances used to test rules 1, 2 and 3

Items transported	Type of category	The container	Observation
CheeseItem01	cheese	Container01	Observation01/observation02
CheeseItem02	cheese	Container01	Observation01/observation02
CheeseItem03	cheese	Container01	Observation01/observation02
CheeseItem04	cheese	Container02	Observation03
CheeseItem05	cheese	Container02	Observation03
CheeseItem06	cheese	Container03	observation04
CheeseItem07	cheese	Container03	observation0

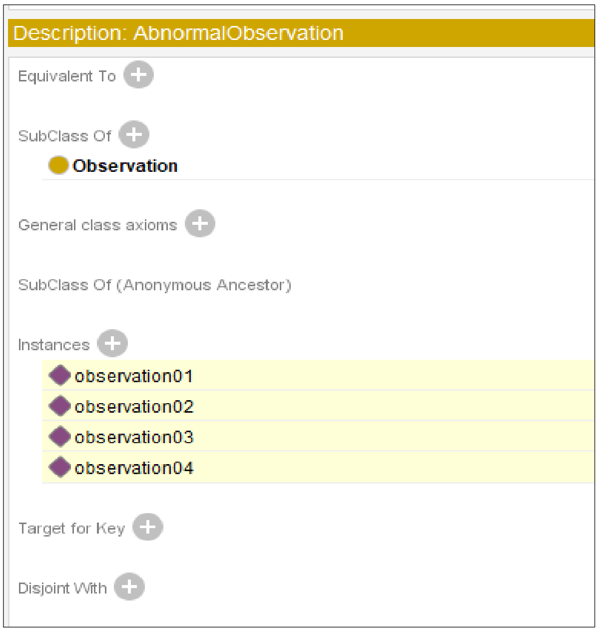
Table 12: The temperature values detected during the observations made in scenario 1

Observation	Measured temperature value
Observation01	7
Observation02	8
Observation03	13
Observation04	-4

Rule 1 deals with the case where the detected temperature values may cause a risk or deterioration in the quality of the products. It corresponds to the individuals (?ob) of the class related to the observations concerning the temperature detected in the containers of the cheese products. If it is between 4 and 10, the observation is declared as abnormal and the product is of type ProductAtRisk.

For the second rule, we check that the temperature value detected is higher than 10. If this is the case, the observation is declared abnormal and the product is considered to be of the deteriorated_product type; that is, the product has deteriorated and its quality could present a health risk to consumers. In the third rule, we check that the average value is lower than -1 to classify it also in the abnormal observation and the product in deteriorated_product.

As illustrated in Figure 17 and Figure 18, temperatures detected in the three containers "Container01", "Container02" and "Container03" generated the triggering of the three rules we described above. And this has led to the inference of new individuals in "AbnormalObservation" class, "productAtRisque" class and "deteriorated_product" classes. Indeed, the detected observations are considered as abnormal observations which present a risk on the product quality. For the products, they are divided into two parts: the products in a risk situation "productAtRisk" which requires an action to prevent them from deteriorating. On the other hand, products those are already deteriorated "deteriorated_product" and should be excluded from the set of products to be transported to the final recipients.



Description: AbnormalObservation

Equivalent To

SubClass Of
Observation

General class axioms

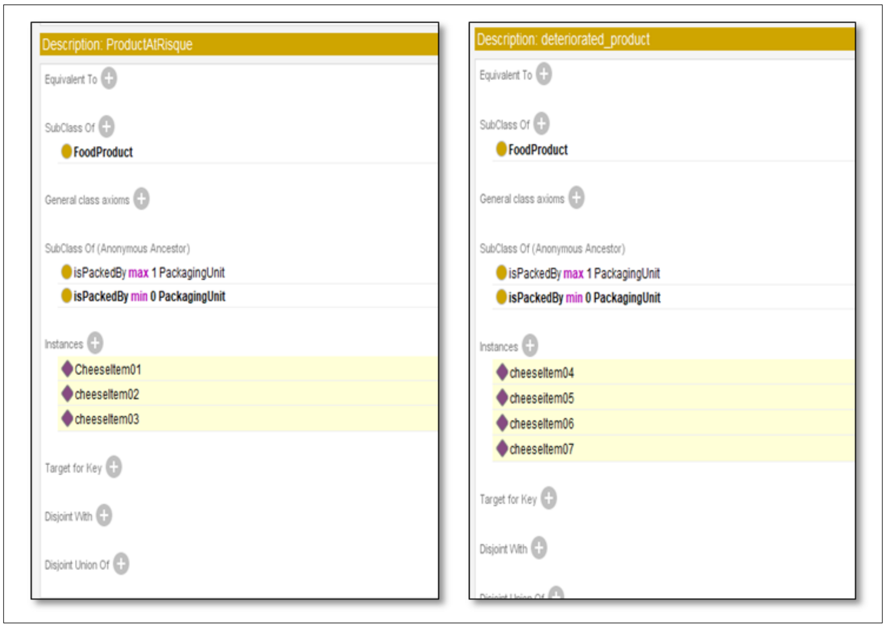
SubClass Of (Anonymous Ancestor)

Instances
observation01
observation02
observation03
observation04

Target for Key

Disjoint With

Figure 17: The individuals inferred in the 'AbnormalObservation' class after executing the rules



Description: ProductAtRisque

Equivalent To

SubClass Of
FoodProduct

General class axioms

SubClass Of (Anonymous Ancestor)
isPackedBy max 1 PackagingUnit
isPackedBy min 0 PackagingUnit

Instances
CheeseItem01
cheeseItem02
cheeseItem03

Target for Key

Disjoint With

Disjoint Union Of

Description: deteriorated_product

Equivalent To

SubClass Of
FoodProduct

General class axioms

SubClass Of (Anonymous Ancestor)
isPackedBy max 1 PackagingUnit
isPackedBy min 0 PackagingUnit

Instances
cheeseItem04
cheeseitem05
cheeseItem06
cheeseItem07

Target for Key

Disjoint With

Figure 18: The individuals inferred in the 'ProductAtRisk' and 'deteriorated_product' classes after running the rules

Using an understandable standard to share observations about products and equipment in outbound logistics processes can help improve the quality and safety of products and the environment involved. Our ontology approach helps companies gain control and global visibility of the SC. It allows tracking of the products and equipment involved to ensure that the consumer receives the products in the right conditions. Furthermore, by using inference capacities of ontologies and semantics rules, we have established a base of appropriate actions to be taken in case of abnormal situations. Our model differs from others by its ability to encompass a description of outbound logistics processes as well as the various concepts used on IOT to ensure a global vision of the transition and changes that may undergo to products and their environment throughout the SC. Additionally, the constructed OLP-IOT can be the basis for building a collaborative architecture for an interoperable sharing and communication between all actors involved in the SC.

7. Managerial implications

The proposed approach provides managerial contribution to improve the competitiveness and success of companies. Primarily, it permits to cope with semantic communication problems in the outbound logistics by offering a common standard to describe and interpret the concepts involved and data generated from the heterogeneous entities and IOT devices. In the same way, this standard allows understanding the activities and events influencing the product lifecycle and helps stakeholders to trace and monitor the status of products among the logistic processes (Tran-Dang et al., 2020). Reliable information could be shared in a timely manner and this can help create a better environment for real-time data exchange, real-time responsiveness, real-time collaboration, real-time synchronization and real-time visibility (Symeonaki et al., 2022a). From the managerial perspective, our approach provides for companies an appropriate solution to predict the possible risks, to ensure control of physical flows and to guarantee the quality and safety of products, especially when dealing with food or pharmaceutical SC that have specific requirements. As a consequence, by using these approaches, companies can increase the level of customer satisfaction and guarantee time and cost saving.

8. Conclusion

Collaboration and information sharing between different stakeholders are key factors for the success and improvement of business performance. This creates the need for standards and formalizations to structure information from different sources in the SC. In this regard, this paper proposes an ontology-based approach that formalizes and models the knowledge of outbound logistics, as well as the data generated by IOT sources involved (sensors, actuators, RFID tags). The ontology is called OLP-IOT, and aims to provide a common semantic framework for the different stakeholders involved in the control, monitoring and visibility of the IOT-based outbound logistics. Accordingly, this approach has been developed based on the NeOn methodology. This methodology takes into account various key design criteria, including reuse of existing standards, modularity and extensibility and expressiveness. Our approach combined and extended existing ontological and non-ontological resources while meeting our requirements. Our model consists of three ontologies; IOT4Log, OLP-Onto and OLP-IOT. OLP-IOT represents the global ontology that allows the complete semantic description of an integrated IOT system in the outbound logistics domain. It represents the result of the linkage of the two ontologies we have developed: IOT4Log and OLP-Onto. Moreover, we have used the proposed ontologies and the Semantic Web Rule Language (SWRL) to establish the rules required to provide high visibility of products and their status and to increase accuracy in the outbound logistics domain. The findings will allow capturing dynamic changes and knowledge evolution over time, such as different normal or abnormal situations that can happen for products, logistic objects or their environment. It also permits to infer high level information that reflects the knowledge of the experts to deduce new knowledge which can help the SC to predict events and react in case of abnormal situations by notifying users or triggering alarms or appropriate actions. The ontological approach has been checked against anomalies and inconsistencies using the Pellet reasoner and has been found to be consistent. Similarly, the proposed approach is evaluated against different ontology evaluation criteria such as structure, usability, skill questions, SPARQL queries and inference rules using SWRL.

The developed model has responded well to the existing gaps. It addressed the needs we specified in terms of improving semantic interoperability and ensuring visibility and control in the SC, however, it still has some limitations. First, a limited number of SC actors were involved in the development of our model; we mainly focused on the analysis of the literature review and studies performed in this context. Second, the proposed model was not deployed in a real SC environment; we just focused on its implementation in a simulation environment. Deploying this model in a real SC will be of great importance in future work.

For future work, we aim to leverage the inference capabilities of ontologies by adding more semantic rules to build a knowledge base that can help control the physical and informational logistics flow. We also intend to exploit our OLP-IOT ontology approach to design a semantic middleware based on event-driven service-oriented architecture for IOT logistics applications. This middleware will ensure proper information dissemination in the distributed logistics environment and flexible data acquisition in addition to near real-time processing of data generated by sensors and various IOT devices.

References

Agarwal, R., Fernandez, D. G., Elsaleh, T., Gyrard, A., Lanza, J., Sanchez, L., Georgantas, N., & Issarny, V. (2016). Unified IOT ontology to enable interoperability and federation of testbeds. *2016 IEEE 3rd World Forum on Internet of Things (WF-IOT)*, 70-75. https://doi.org/10.1109/WF-IOT.2016.7845470

Ahmed, S., Kalsoom, T., Ramzan, N., Pervez, Z., Azmat, M., Zeb, B., & Ur Rehman, M. (2021). Towards Supply Chain Visibility Using Internet of Things: A Dyadic Analysis Review. *Sensors*, *21*(12), 4158. https://doi.org/10.3390/s21124158

Ben-Daya, M., Hassini, E., & Bahroun, Z. (2019). Internet of things and supply chain management: A literature review. *International Journal of Production Research*, *57*(15-16), 4719-4742. https://doi.org/10.1080/00207543.2017.1402140

Benkhaled, S., Hemam, M., Djezzar, M., & Maimour, M. (2022). An Ontology – based Contextual Approach for Cross-domain Applications in Internet of Things. *Informatica*, *46*(5). https://doi.org/10.31449/inf.v46i5.3627

Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., & Taylor, K. (2017). IOT-Lite: A lightweight semantic model for the internet of things and its use with dynamic semantics. *Personal and Ubiquitous Computing*, *21*(3), 475-487. https://doi.org/10.1007/s00779-017-1010-8

Brecht, J. K., Sargent, S. A., Brecht, P. E., Saenz, J., & Rodowick, L. (2019). Protecting Perishable Foods During Transport by Truck and Rail. *EDIS*, *2019*(2). https://doi.org/10.32473/edis-hs1328-2019

Butner, K. (2010). The smarter supply chain of the future. *Strategy & Leadership*, *38*(1), 22-31. https://doi.org/10.1108/10878571011009859

Cao, S., Bryceson, K., & Hine, D. (2020). Improving supply chain risk visibility and communication with a multi-view risk ontology. *Supply Chain Forum: An International Journal*, *21*(1), 1-15. https://doi.org/10.1080/16258312.2020.1717990

Cheng, H., Zeng, P., Xue, L., Shi, Z., Wang, P., & Yu, H. (2016). Manufacturing Ontology Development Based on Industry 4.0 Demonstration Production Line. *2016 Third International Conference on Trustworthy Systems and Their Applications (TSA)*, 42-47. https://doi.org/10.1109/TSA.2016.17

Choy, K. L., Ho, G. T. S., & Lee, C. K. H. (2017). A RFID-based storage assignment system for enhancing the efficiency of order picking. *Journal of Intelligent Manufacturing*, *28*(1), 111-129. https://doi.org/10.1007/s10845-014-0965-9

Christopher, M., & Holweg, M. (2011). "Supply Chain 2.0" : Managing supply chains in the era of turbulence. *International Journal of Physical Distribution & Logistics Management*, *41*(1), 63-82. https://doi.org/10.1108/09600031111101439

Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W. D., Le Phuoc, D., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., … Taylor, K. (2012). The SSN ontology of the W3C semantic sensor network incubator group. *Journal of Web Semantics*, *17*, 25-32. https://doi.org/10.1016/j.websem.2012.05.003

Daly, M., Grow, F., Peterson, M., Rhodes, J., & Nagel, R. L. (2015). Development of an automated ontology generator for analyzing customer concerns. *2015 Systems and Information Engineering Design Symposium*, 85-90. https://doi.org/10.1109/SIEDS.2015.7117017

Daniele, L., & Pires, L. F. (2014a). An Ontological Approach to Logistics. In M. Zelm, M. van Sinderen, L. F. Pires, & G. Doumeingts (Éds.), *Enterprise Interoperability* (p. 199-213). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118846995.ch21

Daniele, L., & Pires, L. F. (2014b). An Ontological Approach to Logistics. In M. Zelm, M. van Sinderen, L. F. Pires, & G. Doumeingts (Éds.), *Enterprise Interoperability* (p. 199-213). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118846995.ch21

d'Aquin, M., Schlicht, A., Stuckenschmidt, H., & Sabou, M. (2007). Ontology Modularization for Knowledge Selection: Experiments and Evaluations. In R. Wagner, N. Revell, & G. Pernul (Éds.), *Database and Expert Systems Applications* (Vol. 4653, p. 874-883). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74469-6_85

De Nicola, A., & Missikoff, M. (2016). A lightweight methodology for rapid ontology engineering. *Communications of the ACM*, *59*(3), 79-86. https://doi.org/10.1145/2818359

Ding, Y., Jin, M., Li, S., & Feng, D. (2021a). Smart logistics based on the internet of things technology : An overview. *International Journal of Logistics Research and Applications*, *24*(4), 323-345. https://doi.org/10.1080/13675567.2020.1757053

Ding, Y., Jin, M., Li, S., & Feng, D. (2021b). Smart logistics based on the internet of things technology : An overview. *International Journal of Logistics Research and Applications*, *24*(4), 323-345. https://doi.org/10.1080/13675567.2020.1757053

Dooley, D. M., Griffiths, E. J., Gosal, G. S., Buttigieg, P. L., Hoehndorf, R., Lange, M. C., Schriml, L. M., Brinkman, F. S. L., & Hsiao, W. W. L. (2018). FoodOn : A harmonized food ontology to increase global food traceability, quality control and data integration. *Npj Science of Food*, *2*(1), 23. https://doi.org/10.1038/s41538-018-0032-6

Elsaleh, T., Bermudez-Edo, M., Enshaeifar, S., Acton, S. T., Rezvani, R., & Barnaghi, P. (2019). IOT-Stream: A Lightweight Ontology for Internet of Things Data Streams. *2019 Global IOT Summit (GIOTS)*, 1-6. https://doi.org/10.1109/GIOTS.2019.8766367

Elsaleh, T., Enshaeifar, S., Rezvani, R., Acton, S. T., Janeiko, V., & Bermudez-Edo, M. (2020). IOT-Stream: A Lightweight Ontology for Internet of Things Data Streams and Its Use with Data Analytics and Event Detection Services. *Sensors, 20*(4), 953. https://doi.org/10.3390/s20040953

Engel, G., Greiner, T., & Seifert, S. (2018). Ontology-Assisted Engineering of Cyber–Physical Production Systems in the Field of Process Technology. *IEEE Transactions on Industrial Informatics*, *14*(6), 2792-2802. https://doi.org/10.1109/TII.2018.2805320

Fernandez, M., Gomez-Pearez, A., & Juristo, N. (1997). *Methontology : From Ontological Art Towards Ontological Engineering*. 8.

Flores-García, E., Jeong, Y., Liu, S., Wiktorsson, M., & Wang, L. (2022). Enabling industrial internet of things-based digital servitization in smart production logistics. *International Journal of Production Research*, 1-26. https://doi.org/10.1080/00207543.2022.2081099

Fortineau, V., PavIOT, T., & Lamouri, S. (2013). 5 root concepts for a meta-ontology to model product along its whole lifecycle*. *IFAC Proceedings Volumes*, *46*(7), 47-52. https://doi.org/10.3182/20130522-3-BR-4036.00061

Fraga, A. L., Vegetti, M., & Leone, H. P. (2020). Ontology-based solutions for interoperability among product lifecycle management systems: A systematic literature review. *Journal of Industrial Information Integration*, *20*, 100176. https://doi.org/10.1016/j.jii.2020.100176

Grangel-Gonzalez, I., Halilaj, L., Auer, S., Lohmann, S., Lange, C., & Collarana, D. (2016). An RDF-based approach for implementing industry 4.0 components with Administration Shells. *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1-8. https://doi.org/10.1109/ETFA.2016.7733503

Helwan University, & Ouf, S. (2021). A Proposed Architecture for Pharmaceutical Supply Chain Based Semantic Blockchain. *International Journal of Intelligent Engineering and Systems*, *14*(3), 31-42. https://doi.org/10.22266/ijies2021.0630.04

Hooi, Y. K., Hassan, M. F., & Ci, T. X. (2012). *Interoperation of Elements in Process Safety Management Via Ontology-Oriented Architecture*. 5.

Jachimczyk, B., Tkaczyk, R., PIOTrowski, T., Johansson, S., & Kulesza, W. (2021). IOT-based Dairy Supply Chain—An Ontological Approach. *Elektronika Ir Elektrotechnika*, *27*(1), 71-83. https://doi.org/10.5755/j02.eie.27612

Janowicz, K., Haller, A., Cox, S. J. D., Le Phuoc, D., & Lefrançois, M. (2018). SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, *56*, 1-10. https://doi.org/10.1016/j.websem.2018.06.003

Janowicz, K., Haller, A., Cox, S. J. D., Le Phuoc, D., & Lefrançois, M. (2019). SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, *56*, 1-10. https://doi.org/10.1016/j.websem.2018.06.003

Jaskó, S., Skrop, A., Holczinger, T., Chován, T., & Abonyi, J. (2020). Development of manufacturing execution systems in accordance with Industry 4.0 requirements: A review of standard- and ontology-based methodologies and tools. *Computers in Industry*, *123*, 103300. https://doi.org/10.1016/j.compind.2020.103300

Keller, R. M. (2016). Ontologies for aviation data management. *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 1-9. https://doi.org/10.1109/DASC.2016.7777971

Kourtis, G., Kavakli, E., & Sakellariou, R. (2019). A Rule-Based Approach Founded on Description Logics for Industry 4.0 Smart Factories. *IEEE Transactions on Industrial Informatics*, *15*(9), 4888-4899. https://doi.org/10.1109/TII.2019.2916622

Lantow, B. (2016). *OntoMetrics : Application of on-line Ontology Metric Calculation*. 12.

Lian, P., Park, D.-W., & Kwon, H.-C. (2007). Design of Logistics Ontology for Semantic Representing of Situation in Logistics. *Second Workshop on Digital Media and Its Application in Museum & Heritages (DMAMH 2007)*, 432-437. https://doi.org/10.1109/DMAMH.2007.52

Lu, Y., Panetto, H., Ni, Y., & Gu, X. (2013). Ontology alignment for networked enterprise information system interoperability in supply chain environment. *International Journal of Computer Integrated Manufacturing*, *26*(1-2), 140-151. https://doi.org/10.1080/0951192X.2012.681917

Martin Placek & ,. (2022). *Industrial Internet of Things (IIOT) market size worldwide from 2020 to 2028*. https://www.statista.com/statistics/611004/global-industrial-internet-of-things-market-size/

Panetto, H., Dassisti, M., & Tursi, A. (2012). ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment. *Advanced Engineering Informatics*, *26*(2), 334-348. https://doi.org/10.1016/j.aei.2011.12.002

Park, D.-W., Lee, G.-T., & Kwon, H.-C. (2008). Ontological Approach to Integration of Event-Centric Logistics Information into EPC Network. *2008 IEEE International Workshop on Semantic Computing and Applications*, 1-8. https://doi.org/10.1109/IWSCA.2008.30

Pizzuti, T., Mirabelli, G., Grasso, G., & Paldino, G. (2017a). MESCO (MEat Supply Chain Ontology) : An ontology for supporting traceability in the meat supply chain. *Food Control*, *72,* 123-133. https://doi.org/10.1016/j.foodcont.2016.07.038

Pizzuti, T., Mirabelli, G., Grasso, G., & Paldino, G. (2017b). MESCO (MEat Supply Chain Ontology) : An ontology for supporting traceability in the meat supply chain. *Food Control*, *72,* 123-133. https://doi.org/10.1016/j.foodcont.2016.07.038

Poveda-Villalón, M., Suárez-Figueroa, M. C., & Gómez-Pérez, A. (2012). Validating Ontologies with OOPS! In A. ten Teije, J. Völker, S. Handschuh, H. Stuckenschmidt, M. d'Acquin, A. Nikolov, N. Aussenac-Gilles, & N. Hernandez (Éds.), *Knowledge Engineering and Knowledge Management* (Vol. 7603, p. 267-281). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-33876-2_24

Rahman, H., & Hussain, I. (2018). *LiO-IOT: A Light-weight Ontology to provide Semantic Interoperability in Internet of Things*. 6.

Rejeb, A., Simske, S., Rejeb, K., Treiblmaier, H., & Zailani, S. (2020). Internet of Things research in supply chain management and logistics: A bibliometric analysis. *Internet of Things*, *12*, 100318. https://doi.org/10.1016/j.IOT.2020.100318

Rozsa, V., Denisczwicz, M., Dutra, M., & Ghodous, P. (2016). *An Application Domain-Based Taxonomy for IOT Sensors*. 11.

Santos, G., Silva, F., Teixeira, B., Vale, Z., & Pinto, T. (2018). Power Systems Simulation Using Ontologies to Enable the Interoperability of Multi-Agent Systems. *2018 Power Systems Computation Conference (PSCC)*, 1-7. https://doi.org/10.23919/PSCC.2018.8442888

Seydoux, N., Alaya, M. B., Hernandez, N., Monteil, T., & Haemmerlé, O. (2015). *Sémantique et Internet des objets : D'un état de l'art à une ontologie modulaire*. 13.

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. https://doi.org/10.1108/09576059710815716

Suárez-Figueroa, M. C., Gómez-Pérez, A., & Fernández-López, M. (2012a). The NeOn Methodology for Ontology Engineering. In M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, & A. Gangemi (Éds.), *Ontology Engineering in a Networked World* (p. 9-34). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-24794-1_2

Suárez-Figueroa, M. C., Gómez-Pérez, A., & Fernández-López, M. (2012b). The NeOn Methodology for Ontology Engineering. In M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, & A. Gangemi (Éds.), *Ontology Engineering in a Networked World* (p. 9-34). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-24794-1_2

Sun, S., Zheng, X., Villalba-Díez, J., & Ordieres-Meré, J. (2020). Data Handling in Industry 4.0: Interoperability Based on Distributed Ledger Technology. *Sensors*, *20*(11), 3046. https://doi.org/10.3390/s20113046

Sure, Y., Staab, S., & Studer, R. (2009). Ontology Engineering Methodology. In S. Staab & R. Studer (Éds.), *Handbook on Ontologies* (p. 135-152). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-92673-3_6

Swar, A., Khoriba, G., & Belal, M. (2022). A unified ontology-based data integration approach for the internet of things. *International Journal of Electrical and Computer Engineering (IJECE)*, *12*(2), 2097. https://doi.org/10.11591/ijece.v12i2.pp2097-2107

Symeonaki, E., Arvanitis, K. G., Piromalis, D., Tseles, D., & Balafoutis, A. T. (2022a). Ontology-Based IOT Middleware Approach for Smart Livestock Farming toward Agriculture 4.0 : A Case Study for Controlling Thermal Environment in a Pig Facility. *Agronomy*, *12*(3), 750. https://doi.org/10.3390/agronomy12030750

Symeonaki, E., Arvanitis, K. G., Piromalis, D., Tseles, D., & Balafoutis, A. T. (2022b). Ontology-Based IOT Middleware Approach for Smart Livestock Farming toward Agriculture 4.0: A Case Study for Controlling Thermal Environment in a Pig Facility. *Agronomy*, *12*(3), 750. https://doi.org/10.3390/agronomy12030750

Tamrakar, A. K., Shukla, A., Kalifullah, A. H., Reegu, F. A., & Shukla, K. (2022). Extended review on internet of things (IOT) and its characterisation. *International Journal of Health Sciences*, 8490-8500. https://doi.org/10.53730/ijhs.v6nS2.7177

Tran-Dang, H., Krommenacker, N., Charpentier, P., & Kim, D.-S. (2020). The Internet of Things for Logistics: Perspectives, Application Review, and Challenges. *IETE Technical Review*, 1-29. https://doi.org/10.1080/02564602.2020.1827308

Tursi, A., Panetto, H., Morel, G., & Dassisti, M. (2009). Ontological approach for products-centric information system interoperability in networked manufacturing enterprises. *Annual Reviews in Control*, *33*(2), 238-245. https://doi.org/10.1016/j.arcontrol.2009.05.003

Uschold, M., & King, M. (1995). *Towards a Methodology for Building Ontologies*. 15.

Verdouw, C. N., Wolfert, J., Beulens, A. J. M., & Rialland, A. (2016a). Virtualization of food supply chains with the internet of things. *Journal of Food Engineering*, *176*, 128-136. https://doi.org/10.1016/j.jfoodeng.2015.11.009

Verdouw, C. N., Wolfert, J., Beulens, A. J. M., & Rialland, A. (2016b). Virtualization of food supply chains with the internet of things. *Journal of Food Engineering*, *176*, 128-136. https://doi.org/10.1016/j.jfoodeng.2015.11.009

Vinoth, P., & Sankar, P. (2016). *Bringing Intelligent Inferences through a Semantic Structure Markup System with the Support of Chemical Ontologies*. 6.

Wang, Y., Huang, L., & vanden Broucke, S. (2019). An activity theory based approach for ontological modelling of collaborative logistics process dynamics. *International Journal of Logistics Research and Applications*, *22*(4), 371-392. https://doi.org/10.1080/13675567.2018.1535650

Yang, H., Kumara, S., Bukkapatnam, S. T. S., & Tsung, F. (2019). The internet of things for smart manufacturing : A review. *IISE Transactions*, *51*(11), 1190-1216. https://doi.org/10.1080/24725854.2018.1555383

Yuen, J. S. M., Choy, K. L., Lam, H. Y., & Tsang, Y. P. (2018). An Intelligent-Internet of Things (IOT) Outbound Logistics Knowledge Management System for Handling Temperature Sensitive Products: *International Journal of Knowledge and Systems Science*, *9*(1), 23-40. https://doi.org/10.4018/IJKSS.2018010102

Zhang, C., Zhou, G., & Lu, Q. (2017). Decision support oriented ontological modeling of product knowledge. *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 39-43. https://doi.org/10.1109/ITNEC.2017.8284816

Zheng, Y., Tetik, M., Törmä, S., Peltokorpi, A., & Seppänen, O. (2020, octobre 14). *A Shared Ontology for Logistics Information Management in the Construction Industry*. 37th International Symposium on Automation and Robotics in Construction, Kitakyushu, Japan. https://doi.org/10.22260/ISARC2020/0175

Zhong, S., Fang, Z., Zhu, M., & Huang, Q. (2017). A geo-ontology-based approach to decision-making in emergency management of meteorological disasters. *Natural Hazards*, *89*(2), 531-554. https://doi.org/10.1007/s11069-017-2979-z

Zhou, H., Benton, W. C., Schilling, D. A., & Milligan, G. W. (2011). Supply Chain Integration and the SCOR Model: Supply Chain Integration and SCOR Model. *Journal of Business Logistics*, *32*(4), 332-344. https://doi.org/10.1111/j.0000-0000.2011.01029.x