View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Archivio istituzionale della ricerca - Politecnico di Milano

Requirements and languages for the semantic representation of manufacturing systems

Elisa Negri[1]*, Luca Fumagalli[1], Marco Garetti[1], Letizia Tanca[2]
[1] *Department of Management, Economics and Industrial Engineering (DIG), Politecnico di Milano, Italy*
[2] *Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Italy*

**Corresponding author: Elisa Negri*
e-mail: elisa.negri@polimi.it

Abstract:

In the last years, attention has been devoted to the development of ontologies, which are ICT conceptual models allowing a formal and shared representation of a particular domain of discourse, and to the use of these representations in a variety of contexts, among which also the industrial engineering can be counted. Within the industrial engineering field, the manufacturing domain has not yet seen a wide application of ontologies. This paper firstly shows the use of ontologies for the semantic annotation of a Web Service–based architecture for the control of manufacturing systems; and then contributes to the research field of manufacturing domain ontologies by proposing a thorough literature review and analysis of the available languages supporting such objective. The paper collects the main requirements that semantic languages must meet to be used in the manufacturing domain with the outlined purpose. In fact, the available semantic languages are several and characterized by different features: the paper identifies the most proper ones for the manufacturing domain representation thanks to their analysis against the main requirements. Lastly, the paper shows how the discussed topics are declined in a real industrial example.

Keywords: Manufacturing system; Semantic languages requirement; Manufacturing domain ontology; Flexibility; Re-configurability

1. Introduction

Current market conditions require companies to be highly flexible to remain competitive on a global scale. Flexibility is the key to face the more educated and demanding customers that ask for quicker delivery, higher variety and more customized products [1]–[4]. In particular, one of the levers to achieve higher flexibility is to have manufacturing systems that are reconfigurable with reasonable time and cost efforts, in order to produce new generations of products [5]. Higher levels of re-configurability require an effort for the development of better conceptual models of the manufacturing domain. A promising direction to this aim could be the development of a manufacturing domain ontology as explained in section 1.2. In fact, ontologies, as a way to model conceptually and logically a system, have been widely proposed and exploited also in other industrial engineering fields [6] and in general engineering was among the earliest fields that applied ontologies [7].

1.1 Research statement

Conceptual models and ontologies can be developed basing on different languages, each with its own characteristics and limits, that are available nowadays [8]. In order to start modelling the manufacturing systems domain, one of the first steps is the selection of a precise and proper language.

After having motivated the industrial interest for ontologies of the manufacturing domain, the first aim of this paper is to investigate what are the requirements for the selection of the proper language for the representation of the manufacturing domain and to put them in a proper framework.

Then, the paper will briefly illustrate the features of the available semantic languages to evaluate them against the identified requirements in the framework.

Eventually, a real case is presented that reflects the industrial relevance of the abovementioned approach and framework and shows the importance of the role of ontologies to face the requirements.

This is also reflected in the structure of the paper: sections 1.2 and 1.3 motivate the industrial interest for manufacturing domain ontologies and review the current state of the art in this research field; section 2 illustrates the requirements that must be met by the semantic languages for the representation of the manufacturing domain; section 3 reviews the available semantic languages and discusses the matching between them and the identified requirements; section 4 shows how the role of ontological modelling and the identified requirements are declined in a real industrial case and Section 5 is dedicated to the concluding remarks and suggestions for future work.

1.2 The role of ontological modelling in the manufacturing domain

Despite the fact that a high level of flexibility is reached at the mechanical level, the re-configurability level of the control systems is still poor [9]. It has been estimated by Colombo that 70% of the engineering teams' effort is directed to modify the control system when a new machine is introduced in the production system [10].

Literature suggests that a possible answer to the issues related to control architecture flexibility and re-configurability at software level is the use of a distributed control architecture, based on Cyber Physical Systems, smart components that are put into communication thanks to well-established standards such as Profibus, or into a Service Oriented Architecture (SOA) [11]–[15]. In particular, the SOA architecture offers the potential for device interoperability, thanks to its features of message-based communication, loose coupling and open standards. Such a control architecture encapsulates the manufacturing processes in services (namely, Web Services) that are offered on a Web-based communication network, where the control system may find them and invoke them through the orchestration and choreography mechanisms [16]. These are needed for the composition and execution of the services related to the manufacturing processes in the proper sequence [10]. In this way, re-configurability at the software level is made possible. However, high costs and a long time are still required to implement new configurations. In fact, changes in the physical manufacturing system must correspond to modifications in the control software by human programmers. This is due to the lack of a machine-readable semantic description of the system and of the operations to be performed in the specific context of the manufacturing system at hand: therefore, semantics is still interpreted by the human programmer who will include the necessary changes into the control software. A possible solution to this issue could be the development of a proper semantic model of the production system and make it accessible to the control software through the use of semantically-enriched Web Services (i.e. Semantic Web Services). Within such an approach, human interventions are no more needed, or only limited to a very small extent, because the semantics makes the knowledge about the manufacturing system itself understandable to the control software: this opens the way to automatic reconfiguration of the control software in case of physical modifications in the production system [9], [10].

A way to add semantics to Web Services is their annotation with ontological models, that provides a semantic description of the production system and can be exposed as services on a Web-Service based SOA control architecture [17]. According to the definition by Gruber, an ontology is an "explicit specification of a conceptualization", where a conceptualization is an abstract, simplified view of the world that we need to represent for some specific purposes [18].

Ontologies support class-based, or object-oriented, description of a knowledge domain, expressing taxonomies and semantically rich relationships among concepts, supporting information retrieval through reasoning. Moreover, by their nature, distinct ontologies can be integrated by creating "bridge" relationships among some concepts of the different ontologies [19]. This characteristic is particularly useful in the description of complex manufacturing systems.

Already in 1999, Schlenoff understood the potential of ontologies in the manufacturing domain (unambiguous communication, shared terminology and semantic alignment, and industrial information infrastructure in that they provide data in computational form) [20]. The possible uses of ontological representations of the manufacturing domain are not limited to the applications in control architectures, but, as pointed out by Garetti and Fumagalli, they can also support design, simulation, planning and scheduling, performance assessment and data integration in the field [21].

1.3 State of the art on conceptual modelling in manufacturing

Since many years, the topic of conceptual modelling for the manufacturing domain is an open research stream. Some of the first works on this topic date back to the 90s, when early research on conceptual modelling and ontology development of the manufacturing systems was proposed by Politecnico di Milano [22]. The P-PSO, Politecnico di Milano – Production Systems Ontology was proposed as a complete modelling of the manufacturing domains that could be used for information exchange, design, control, simulation and other applications [21], [23].

Since then, many other research groups and research projects have worked on this topic. The success of semantic and conceptual models in the manufacturing domain can be justified by the many characteristics and potentialities of such models. In particular, they are implemented as ontologies that allow sharing the same vocabulary, not relying on human programmers' interpretations of the natural language that can sometimes bring to misunderstandings, according to Guarino [24].

The developed ontologies for the manufacturing domain range from the most general, the so-called foundational ones, to the very specific for a certain context within the more general manufacturing domain. Each of them has its importance, regardless of the detail level they have [25]. Also the motivations that lead to their creation can be different: ontologies applications bring benefits covering automatic re-configurability, interoperability, creation of a common vocabulary, and knowledge sharing and reuse. The various applications in manufacturing differ also on the level of the potentials offered by ontologies: some are simply structured machine-understandable vocabularies of a certain domain, others are built with the purpose of inferring new knowledge starting from the structured information in the model.

The applications of ontologies in the manufacturing domain may depend on various reasons, the main of which are listed below:

- Some claim to use them for the support to *reconfiguration of manufacturing systems* without human intervention; in particular, a reconfiguration agent is based on the ontological knowledge of the manufacturing system [16], [26], [27].
- Colledani et al. [28] conceptually modeled the manufacturing domain perspective on products, processes and production systems in order to *model them in an integrated way*. Other example references for ontologies used as integrated models of manufacturing systems are: [29]–[31].
- In [17] and [32] ontologies are also created that represent the manufacturing domain but with another objective: the *inter-enterprise interoperability*; for this reason, along with classes representing resources and operations, they also inserted enterprise- and strategy-related classes. Also other authors deem ontologies in manufacturing the way to address inter-enterprise interoperability issues: [33], [34].
- The problem of *interoperability among different systems in the enterprise* has been addressed by [35], who propose a development approach for formal ontologies and use it to represent production systems for the interoperability with legacy systems. Also [36] insists on enterprise systems interoperability in manufacturing by building a product ontology. Other examples are: [37], [38].
- *Knowledge sharing* is another motivation arisen in literature to use ontologies in the manufacturing domain. As an example, [39] and [40] used an ontology as a basis for common understanding between manufacturing or assembly engineers and design engineers. This ontology comprised both aspects related to manufacturing (resources, processes, parts and production plan) and related to design (geometrical measures). Another example of ontology used for knowledge sharing is in [41] and [42], where the focus is on gathering all important information about the product lifecycle management (PLM) into an ontological base.
- Connected to knowledge sharing is *knowledge reuse,* which also is one of the reasons ontologies are developed and is at the basis of *interoperability among different technical products* manufactured by different vendors, the benefit of this is to reuse the same ontology or knowledge contained in different specific applications that would otherwise require the building of a new knowledge structure [43], [44]. Long [45] gives an example of how to exploit the potential knowledge reuse in the context of Manufacturing Execution Systems that control the production processes. [46] focuses in the knowledge sharing and re-use for the production systems of aerospace composites.
- Also the *inference capability* of ontologies might be the reason why they are created. It is the case in [47], trying to exploit this capability of ontologies to allow the engineers to define as few features as possible, making the ontology infer the rest.

Even if ontologies have proven a promising approach to achieve many benefits in industrial engineering, they still have very limited application and are not fully deployed in commercial tools for the industrial manufacturing practice. In order to facilitate the development of proper manufacturing domain ontologies to be exploited in the industrial reality, the current paper proposes a way to choose the most appropriate ontology language starting from an identified framework of the requirements of the manufacturing domain, keeping in mind what outlined in section 1.2.

2. Requirements from the manufacturing systems domain

This paper wants to contribute to the discussion present in literature about the requirements in the selection of the ontological language [48], [49], and in particular related to the representation of manufacturing-systems domain ontologies. The authors' perspective is the one of manufacturing systems domain experts who are interested in ontological languages requirements corresponding to practical domain requirements coming from the industrial production field, having in mind the ultimate purpose for the development of the manufacturing domain ontology, that is the semantic annotation of web-service-enabled control architectures of manufacturing systems.

It is important to recognize that the requirement levels for the production systems are several. The current work will only focus on those that are fulfilled through the proper selection of the ontological modelling language; while those that are more related to the quality of the conceptual representation (e.g. the possibility to capture different variants of production processes) will not be discussed in this paper.

Basing on a thorough literature review, that started with the earliest works on this topic until the recent publications, it has been clear that languages for the conceptual and semantic modelling of the manufacturing domain should ensure four main requirements: (i) they must allow conceptual modelling and data storage, (ii) they must offer easy use and maintenance of the model, (iii) they must support interoperability, and (iv) they must support automated reasoning.

A short explanation of each of them is given in the following lines. In this description each of the main requirements is subdivided into elementary requirements, that allow the achievement of the main requirement, as shown in Figure 1.

i. The first requirement is about the possibility to use the ontology as a **conceptual model of the manufacturing domain and to store data in it**: in particular, this has to do with the selection of an appropriate ontological language that must give the possibility to build the knowledge base [50], and must be general enough and adaptable to describe different production systems [28], [32]. Moreover, the storage capability should be persistent (information should not be lost in the case of a system crash), for this reason a secure and persistent knowledge base must support the language and used tools [51].

ii. The second requirement deals with **ease of use and maintenance** of the model. To ensure this, the ontology must have a compact syntax that is intuitive to humans: it is even better if a graphical notation is provided to help human reading and if consistency is assessed against the human knowledge representation [50]. Moreover, the representation should be object-oriented, because this allows for abstract classifications (describing only interesting details of the objects, ignoring the minor ones), to encapsulate details (hiding unnecessary details), to build modularity (elements of the model highly decoupled but consistent), and to aggregate objects (an object composed of other objects) [28]. In order to support easy updates, the description should be made at different levels of detail, where each level of detail should be easily extended if needed; in other words, it should be scalable and extensible [28].

iii. The third requirement touches the **interoperability issue** that is one of the earliest arisen and most felt by researchers and practitioners working with ontologies [52]. Interoperability problems concern compatibility of data semantics and representation, software applications, communication paradigms and system architectures; while all the latter aspects can be solved by building appropriate language and protocol conversions, the semantic interoperability issue is much felt in production environments because manufacturing engineering software applications sometimes associate different meanings to the same terms making it particularly difficult to exchange process information. A solution for the semantic interoperability problem is the use of a proper ontology for the manufacturing domain [32].. An obvious and necessary requirement for semantic interoperability is making the ontological model machine-understandable [50]. The developed ontology also should be able to bridge to other ontological models representing different aspects related to each other (physical system, product, process, control…), in this way providing an integrated description of processes and products, this ability is usually called "matching" and "mapping" of different ontologies [19], some of which might be already existing and not created ad-hoc. In addition it should include both universal and domain categories: reasoning on universal categories (such as time and space) should be allowed. To this aim, the entities should be accordingly specified, for example an object has a time and location, so there will not be a category that defines time or spatial concepts. Moreover, if the ontology has to be exploited in a web service control architecture, as envisioned by the ontologies use paradigm illustrated in section 1.2, the ontology must have a proper link with existing web standards to ensure semantic interoperability [50].

iv. The fourth requirement states that it should **support the automated reasoning**. This is firstly achieved with the use of a well-defined formal semantics [50]; moreover, the chosen ontology language should present some reasoning properties that make the ontological model not only a mere specific domain vocabulary but a model proper also for automated reasoning [50]. It should also be ensured that the reasoning time is finite and, possibly, efficient, because in real word implementations it is not acceptable that the answer to knowledge inference does not come within a fixed time, that is compatible with the application runtime in which the ontology is used [53]. The modelling should be able to express all objects in the manufacturing system with the possibility to describe an object as a sub-object of another object that has already been defined; this property is called "inheritance", because the sub-object will inherit all the characteristics of the object above, just adding more detailed characteristics that are specific to it[54]. For the exploitation in a web service control architecture, such as the one envisioned in section 1.2, services must be able to access the information stored in the ontological knowledge base and to update it through queries: this elementary requirement could be named "dialog with services" [50].

Figure 1 shows a representation of the four main requirements of the ontological language for the manufacturing domain, with a summarizing list of the illustrated elementary requirements for each of the main requirements.

3. Languages review and assessment

The review of the existing conceptual and semantic languages started with a literature search based on the following keywords: semantic languages, manufacturing knowledge, knowledge based systems, SOA, Web Services, Semantic Web Services, ontology, interoperability. With these keywords, a huge number of research papers can be collected. [55] has proposed a distinction in research streams to classify the possible types of research work carried out in this context. In order to clarify what type of paper have been considered for the analysis that follows, this paper refers to the third main research stream identified in [55] in the field of semantic languages and conceptual modeling. This stream reviews and compares different semantic languages: either discussing language alternatives for a specific implementation purpose or presenting the historical evolution of semantic languages.

Examples of such a research stream have been provided by [8], [50], [56]–[64].

The analysis of works in this research stream has led to the identification of the available semantic languages and their features. The considered languages in the analysis were the following: KIF, OntoLingua, Operational Conceptual Modelling Language, FrameLogic, Loom, DublinCore, SHOE, XML(S), XOL, RDF(S), OIL, DAML, DAML+OIL, DAML-L, UML, OWL, and the OWL sublanguages: OWL Lite, OWL DL, OWL Full, Context-OWL, OWL-Eu, OWL-E, OWL Flight. Table 1 shows the main features of these languages, taken from the following references: [8], [28], [32], [50], [57], [58], [63]–[65].

As it is clear from Table 1, many semantic languages do exist. Each allows a different expressivity level and a different reasoning. Each of them answers in a different way to the requirements mentioned in Section 2, it is thus necessary to evaluate the available languages with respect to those requirements.

i. *The primary requirement for the semantic language to be used for the manufacturing domain representation is that it supports the "conceptual modelling and data storage" (see Fig. 1).*

This requirement is not fully supported by SHOE language because in fact it has been created to annotate HTML pages, and therefore it has no use for manufacturing domain conceptual models; it must then be removed from the selection of possible semantic languages.

Conceptual modelling is also not fully supported for Dublin Core and OWL Flight, both based on Logic Programming (LP) and not on Description Logics (DL), like the other languages. In fact, contrarily to Description Logics (DL), Logic Programming (LP) provides a limited conceptual modelling (because it is based on "flat" predicates): for this reason languages that are LP-based are not advisable for conceptual modelling. Moreover, also the other requirements presented in section 2 are not fulfilled by LP-based languages:

- If negation constructs are inserted, reasoning becomes undecidable in general, so it is not sure that it is time-finite; (sub requirement of main requirement 4);
- Complexity is not under control even if negation is avoided; therefore they do not offer a compact syntax, against (sub requirement of main requirement 2).

For these reasons, DL-based languages are more appropriate for conceptual modelling, can support negation constructs (that might be needed in manufacturing domain conceptualizations), complexity can be explicitly controlled and thus the reasoning time can be limited. Consequently, Dublin Core and OWL Flight cannot be included in the selection, because they are based on Logic Programming.

ii. *It seems that the second main requirement set out in section 2 "easy usage and maintenance"* is met by all the remaining languages to an acceptable extent.

iii. *The third main requirement from section 2 support to "interoperability"* is not ensured by the following semantic languages: KIF, Ontolingua, OCML, FLogic and Loom because they do not have the proper link with web standards (sub requirement of main requirement 3). Thus, these languages must be removed from the selection of the possible languages envisioned for the use in manufacturing described in section 1.2. In addition, these languages do not support reasoning properties at all or only to a very limited extent, therefore they should be cut out of the selection because they do not meet also the main requirement number 4 "support to reasoning".

iv. The fourth main requirement is meant to ensure "reasoning" of the semantic language.

A number of languages must be excluded from the selection because they do not meet this requirement:

- XML(S), XOL, UML because they do not support reasoning properties (sub requirement of main requirement 4);
- RDF(S) and OWL Full because they do not support time finite reasoning (sub requirement of main requirement 4).

As a conclusion, it can be concluded that the candidate languages that could be used to model the manufacturing domain ontology are OIL, DAML (DAML+OIL and DAML-L), OWL and its sub-languages: OWL Lite, OWL DL, C-OWL, OWL-Eu, OWL-E, because they satisfy all the requirements set out on section 2.

Among these candidate languages, OIL and DAML (DAML+OIL and DAML-L) are not fully advisable, in fact they are built from XML(S) and RDF(S) by means of incremental improvements and have been surpassed by the OWL family. For this reason it can be concluded that the OWL languages, except those that have been excluded during the previous discussion (i.e. OWL Flight and OWL Full), are the most appropriate to model the manufacturing domain, namely:

- OWL Lite,
- OWL DL,
- C-OWL,
- OWL-Eu,
- OWL-E.

According to the specific implementation situation, the choice of the most appropriate language should be done with the awareness that the more powerful the reasoning and the more complex the model, the longer will be the time to perform reasoning, with the risk not to have time-efficient reasoning.

4. Industrial example

An OWL ontology has been implemented in a new control architecture that deploys the ontologies use paradigm illustrated in section 1.2, in order to reach the benefits of overcoming rigidity of the traditional control solutions. This implementation has been part of the activities in the eScop European funded project (www.escop-project.eu).

The industrial context in which it has been developed is a logistic integrator company that considers offering flexible solutions to customers in a highly changing market as a competitive advantage. One of the specific flexibilities that they are interested in achieving is the possibility to integrate elements that come from different vendors into their solutions.

The system on which the open paradigm has been implemented is a picking system, composed by four main subsystems, as shown in Figure 2:

1) a carousel ring, identified with letter A in the picture, which is used to create a buffer of the pallets that are called for picking the most in order to speed up picking operations without too often interfacing with the warehouse;
2) the warehouse station, both for input and output of boxes from and to the main warehouse, identified with letter B;
3) a buffer station also used for picking, letter C;
4) a gravity conveyor to reject non-satisfactory products, letter D.

For such a system, a high level control system is needed that manages the boxes that must be called from and sent back to the main warehouse.

The current control system is based on a rigid hierarchical three-level control architecture, that follows the IEC 62264.

i) The lower level is composed of electro-mechanical devices and controllers that take low level decisions, such as stopping box movement if the next buffer element is already occupied.
ii) The middle level is composed of Device Control Units, such as Programmable Logic Controllers, that act as control stations by coordinating low level devices for the correct sorting of boxes and collecting fragmented and scattered information from the field devices.
iii) The higher level is a supervisory level providing information on orders and keeping the coordination with external systems, such as the warehouse control system.

This current control architecture is deployed in a rigid hardware structure, that is efficient only for stable contexts. The company wanted to shift to a control network that exploits new communication technologies in order to be able to integrate heterogeneous devices into a large distributed network through web services protocols, as envisioned by the ontologies use paradigm illustrated in section 1.2. This is done by exploiting one of the results of the eScop project: the control architecture that is presented in Figure 3. This is composed of five layers: physical, representation, orchestration, visualization and interface layers.

The physical layer includes smart devices connected to the physical production equipment that allow the low level control, being Cyber Physical Systems (CPS), that integrate physical, computational and control elements [11]. These Cyber Physical Systems are in communication on a Web Service network.

The representation layer is composed of the instanced ontological model, representing the knowledge about the system configuration and components, and of the ontology service that allows it to be exposed as a service on the rest of the control architecture. In particular, the ontology provides the information about the production system stored within, through queries that are posed by the ontology service.
The orchestration layer that orchestrates the production activities, exposed as Web Services in the physical layer, basing on information received from the knowledge of the production system stored in the ontology. The ontology acts, therefore as semantic annotation of the production Web Services, that is needed by the orchestrator to control the production environment. The orchestration layer is composed of the service composer and the orchestrator service; the former receiving task needs and the second orchestrating the services in a way to satisfy the needs. This functioning is described by Figure 4, representing in a UML sequence diagram the interactions between the orchestration layer components (service composer and orchestrator service) with the devices of the physical layer and the ontology (as mentioned the ontology service acts as interface of the ontology with the rest of the architecture).
The last two layers, the interface and the visualization, are communicating with the external world: the interface layer communicates with external software applications, while the visualization layer displays the information to people, acting as a human-machine interface.

In order to change the control paradigm of this company, it is necessary to avoid the static encapsulation of the knowledge about the system itself that is now included into the low level control, not allowing easy reconfiguration and interoperability of components from different vendors. The most promising approach to abstract this knowledge (and store it in a way that reconfigurations are allowed quicker and with less efforts) is the use of an ontology that captures the conceptual structure of the domain. In fact, the physical configuration of the production system is flexible, modular and based on Cyber Physical Systems; this allows higher degrees of flexibility, a more rapid reconfiguration and shorter ramp-up times when combined with communication standards such as Web Services. Although Cyber Physical Systems can represent the basis for a modular architecture, they do not have a systemic view of their role inside the production system. For this reason, the knowledge about the specific production system must be provided to the control system that will then coordinate the production processes and activities, according to the specific role of each module inside the system. If the knowledge is provided by an ontological model of the manufacturing domain, instanced on the specific production system and updated according to changes occurring in its configuration, the control system can always rely on the updated knowledge about the production system [43].

The ontology in this system is not simply used as a static vocabulary, but as a dynamic knowledge base where information can be retrieved and updated through SPARQL queries by Web Services that put it in connection with the rest of the control architecture for a successful run-time control of the system.
The manufacturing domain ontology model that has been developed is able to represent the different aspects of a manufacturing system: the physical aspect, the technological aspect and the control aspect. The physical aspect allows to describe the physical components of the system, such as storage compartments, transporters, sensors and processors; the technological aspect describes the technological and transportation routings and their related operations; and the control aspect supports the control system with information on the orders and the picking lists. These aspects are represented in the ontology model, each aspect is composed of a number of classes linked to other classes by association, by inheritance or by aggregation relationships. Figure 5 shows a screenshot of the classes in the ontological model of the manufacturing systems domain that is developed in the mentioned project, eScop. The classes in the picture are in alphabetical order, as the ontology editor shows them. The ontology model has also properties and constraints that are needed to represent the manufacturing systems domain and that are not shown in the picture for space constraints.
When a new configuration of a given system is established, the ontology instance devoted to the specific system configuration is created by the operator starting from the general manufacturing domain ontology model. The instance knowledge base supports the control system in sending commands to the production floor. Flexible command capability is achieved through the configuration knowledge content of the ontology. When the control system interacts with information structured in an ontology, it is possible to flexibly control the entire shop floor because the control architecture, in fact, is flexible to any variation of the system (e.g. the change in the number of storage compartments, namely a change in the number of instances of compartment).
This brief illustration of an industrial example is aimed at showing the fact that the requirements collected in the literature review - that must direct the choice of the language for the semantic representation of manufacturing (and internal logistics, as a subdomain of manufacturing) - are very relevant also in the described industrial case.

i. *The first and second main requirements for the semantic language to be used for the manufacturing domain: support to the conceptual modelling and data storage and easy usage and maintenance (see Fig. 1).* The fact that the model must be at first instanced by an operator reflects the needs dictated by the first two requirements for semantic languages: the need to have a conceptual model that allows the inclusion of stored data; and the need for an easy usage and maintenance by the operator himself that must easily understand it to instance the individual elements of the specific system.
ii. *The third main requirement: support to "interoperability" (see Fig. 1).* One of the very benefits of ontologies in such a described architecture when compared to other possible knowledge bases is the fact that they offer the basis for interoperability at different levels: at software level to support different manufacturing and logistics software applications; at physical level to integrate and make different vendors' element communicate and at

human level to represent and share the conceptual model of the system among the people with different backgrounds that might be collaborating on the same manufacturing system.

iii. *The fourth main requirement: support to "reasoning capabilities" (Fig. 1).* This capability is what makes the run-time possible through automated retrieval of information and automated data update in the instanced ontological model. It is therefore one of the basic functional requirements that must be supported by the chosen semantic language. The inheritance feature is one of the most required by the manufacturing domain model because it is possible to define the characteristics of an object (such as a *machine*) and then to detail it in the different types (*NC machine*,…) without the need to re-specify the common features.

iv. The selected language for this system is the OWL DL language, as defined by W3C, that is built on XML (S) and RDF (S) (http://www.w3.org/2002/07/owl#), because as it was shown in the previous section it meets all the four main requirements for the manufacturing domain and with respect to the other possible languages offers the maximum representation potential.

5. Conclusions

In the last years, more and more attention has been devoted to the development of ontologies, ICT conceptual models allowing formal and shared definition of the types, properties, and interrelationships of the entities that exist for a particular domain of discourse [18], and to the use of these representations in a variety of contexts. Indeed, ontologies have been employed to describe the domain of medicine [66] that of military operations [67], [68] and social sciences [69], just to name a few. Also the industrial engineering has seen many application of ontologies in different phases of the products, services and production processes [6]. In fact, ontologies are a way to represent the knowledge related to a domain and enable the representation of this knowledge within automated and software systems, thanks to their formal nature [18]. This possibility brings enormous benefits into the automated systems applications in which they are used: the paper has focused on those related to the manufacturing domain field. The available semantic languages that can be used in these proposed applications are several, each of them characterized by different reasoning capabilities, complexity, levels of difficulty in programming, and other features [8].

The paper showed a possible use for ontologies that is aimed at overcoming some of the problems related to reaching higher flexibility and re-configurability of the manufacturing systems. That is the addition of a manufacturing domain ontology in Web Service- oriented control architecture of production systems in order to allow easy interoperability and easy re-configuration of the system at software level in case of physical modifications. The ontology acts as a semantic annotation of the Web Services, that allows the control of the production system based on the semantics contained in the ontology.

Keeping in mind the illustrated use of ontologies in the manufacturing domain, the paper concentrated on the choice of the most proper semantic languages for the development of manufacturing domain ontologies: firstly the requirements that must be met by these semantic languages are collected and then a thorough literature review and analysis of the available semantic languages in literature and commerce was carried out. Finally, the assessment of the available languages was performed against the identified requirements.

Final conclusions are that the four main requirements for a semantic language of the manufacturing domain are:

(i) they must support conceptual modelling and data storage,
(ii) they must support easy use and maintenance of the model,
(iii) they must support interoperability,
(iv) they must support automated reasoning.

Against these requirements, some of the semantic languages under consideration were not considered acceptable for the purpose of manufacturing domain ontology development, and in particular: DublinCore, FLogic, KIF, Loom, OCML, OntoLingua, OWL Flight, OWL Full, RDF (S), SHOE, UML, XML (S)and XOL.

Some languages met the requirements but have been surpassed by other languages and therefore, their use is not advised, these are OIL and DAML family.

The languages that are advised for this use are the OWL and the OWL sublanguages: OWL Lite, OWL DL, C-OWL, OWL-Eu and OWL-E. The final language choice depends on the specific implementation constraints, in particular time constraints that must meet the runtime requirements of the specific application.

Regarding future developments of this research work, the future directions can be twofold:

i) New industrial engineering fields have already been identified, where the collection of requirements for semantic languages and the assessment of the most appropriate languages should be performed. One of the first fields that is already starting to be addressed is the process industry domain.

ii) The ontologies use as semantic annotation of a Web Service-based control architecture of manufacturing systems is not the only application in the manufacturing domain (and, in general, in the industrial engineering) that could benefit from an introduction of ontology-based knowledge. Other possible applications should be analysed: as an example, further research is already currently going on investigating the potentialities that ontologies could offer to human-machine interface visualization systems in terms of flexibility and quicker updates.

A further future direction of investigation is also the continuous querying activity presented by Barbieri and colleagues that retrieve information and performs reasoning on RDF datastreams [70], [71]. This may have enormous applications in

production environments where sensor values are updated with a high frequency and a semantic support for such a quick updating and storing of historical data is not yet supported by other ontological languages.
Another future development could further investigate into the possibility to control the production system thanks to a generic orchestrator that takes the knowledge about the system from the instanced ontology of the manufacturing domain.

6. Acknowledgements

This article is a revised and expanded version of a paper entitled "A review of semantic languages for the conceptual modelling of the manufacturing domain", presented at the XIX Summer School "Francesco Turco", held in Senigallia, AN, Italy, from 9th to 12th September 2014.
The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement n° 332946 and from the Italian Ministry of Education, Universities and Research (MIUR), correspondent to the project shortly entitled *eScop, Embedded systems for Service-based control of Open Manufacturing and Process Automation*.

7. References

[1] M. M. Tseng and J. S. Hu, "Mass customization," in *CIRP Encyclopedia of Production Engineering*, Springer Berlin Heidelberg, 2014, pp. 836–843.

[2] S. L. Cohen, "Effective global leadership requires a global mindset," *Ind. Commer. Train.*, vol. 42, no. 1, pp. 3–10, 2010.

[3] F. Salvador and C. Forza, "Configuring products to address the customization-responsiveness squeeze: A survey of management issues and opportunities," *Int. J. Prod. Econ.*, vol. 91, pp. 273–291, 2004.

[4] J. Meredith and U. Akinc, "Characterizing and structuring a new make-to-forecast production strategy," *J. Oper. Manag.*, vol. 25, pp. 623–642, 2007.

[5] Y. Koren and M. Shpitalni, "Design of reconfigurable manufacturing systems," *J. Manuf. Syst.*, vol. 29, no. 4, pp. 130–141, 2010.

[6] V. Fortineau, T. Paviot, and S. Lamouri, "Improving the interoperability of industrial information systems with description logic-based models—The state of the art," *Comput. Ind.*, vol. 64, no. 4, pp. 363–375, May 2013.

[7] X. Ma, J. Bal, and A. Issa, "A fast and economic ontology engineering approach towards improving capability matching: Application to an online engineering collaborative platform," *Comput. Ind.*, vol. 65, no. 9, pp. 1264–1275, 2014.

[8] F. Giunchiglia, F. Farazi, L. Tanca, and R. De Virgilio, "The Semantic Web Languages," in *Semantic Web Information Management*, Springer-Verlag Berlin - Heidelberg, 2010, pp. 25–38.

[9] J. Lastra and I. Delamer, "Semantic web services in factory automation: fundamental insights and research roadmap," *Ind. Informatics, IEEE Trans.*, vol. 2, no. 1, pp. 1–11, 2006.

[10] A. W. Colombo, F. Jammes, H. Smit, R. Harrison, J. L. M. Lastra, and I. M. Delamer, "Service-Oriented Architectures for Collaborative Automation," *Ind. Electron. Soc. 2005. IECON 2005. 31st Annu. Conf. IEEE*, pp. 2649–2654, 2005.

[11] R. Baheti and H. Gill, "Cyber-physical Systems," in *The Impact of Control Technology*, no. 12, 2011, pp. 161–166.

[12] J. Lee, B. Bagheri, and H. Kao, "A Cyber-Physical Systems architecture for Industry 4 . 0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, 2015.

[13] K.-J. Lin and M. Panahi, "A real-time service-oriented framework to support sustainable cyber-physical systems," in *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on. IEEE*, 2010, pp. 15–21.

[14] H. J. La and S. D. Kim, "A service-based approach to designing cyber physical systems," in *Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on. IEEE*, 2010, pp. 895–900.

[15] D. Duggan, "Service-Oriented Architecture," in *Enterprise Software Architecture and Design: Entities, Services, and Resources*, 2012, pp. 207–358.

[16] M. Loskyll, J. Schlick, S. Hodek, L. Ollinger, T. Gerber, and B. Pirvu, "Semantic service discovery and orchestration for manufacturing processes," *Emerg. Technol. Fact. Autom. (ETFA), 2011 IEEE 16th Conf.*, 2011.

[17] M. Cai, W. Y. Zhang, and K. Zhang, "ManuHub: A Semantic Web System for Ontology-Based Service Management in Distributed Manufacturing Environments," *IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans*, vol. 41, no. 3, pp. 574–582, May 2011.

[18] T. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *Int. J. Hum. Comput. Stud.*, vol. 43, pp. 907–928, 1995.

[19] P. Zhan, U. Jayaram, O. Kim, and L. Zhu, "Knowledge representation and ontology mapping methods for product data in engineering applications," *J. Comput. Inf. Sci. Eng.*, vol. 10, no. 2, 2010.

[20] C. Schlenoff, R. Ivester, D. Libes, P. Denno, and S. Szykman, *An Analysis of Existing Ontological Systems for Applications in Manufacturing and Healthcare*. 1999.

[21] M. Garetti and L. Fumagalli, "P-PSO Ontology for Manufacturing Systems," *Inf. Control Probl. Manuf.*, vol. 14, no. 1, pp. 449–456, 2012.

[22] A. Bartolotta, E. Corradi, and M. Garetti, "Developing an ontology for the modelling of manufacturing systems," in *IFIP International Enterprise Modeling Conference*, 1999.

[23] M. Garetti and L. Fumagalli, "Role of ontologies in open automation of manufacturing systems," in *Proceedings of the XVII Summer School in Industrial Mechanical Plants - 12/9/2012-14/9/2012, Venice, Italy*, 2012.

[24] N. Guarino, "Formal Ontology and Information Systems," *Form. Ontol. Inf. Syst. Proc. FOIS'98*, no. June, pp. 3–15, 1998.

[25] S. Borgo and P. Leitão, "The role of foundational ontologies in manufacturing domain applications," *Move to Meaningful Internet Syst. 2004 CoopIS, DOA, ODBASE*, pp. 670–688, 2004.

[26] Y. Alsafi and V. Vyatkin, "Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing," *Robot. Comput. Integr. Manuf.*, vol. 26, no. 4, pp. 381–391, 2010.

[27] M. Garetti, L. Fumagalli, a. Lobov, and J. L. Martinez Lastra, "Open automation of manufacturing systems through integration of ontology and web services," *IFAC Proc. Vol.*, pp. 198–203, 2013.

[28] M. Colledani, W. Terkaj, T. Tolio, and M. Tomasella, "Development of a Conceptual Reference Framework to Manage Manufacturing Knowledge Related to Products, Processes and Production Systems," in *Methods and Tools for Effective Knowledge Life-Cycle-Management*, A. Bernard and S. Tichkiewitch, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 259–284.

[29] A. Giovannini, A. Aubry, H. Panetto, M. Dassisti, and H. El Haouzi, "Ontology-based system for supporting manufacturing sustainability," *Annu. Rev. Control*, vol. 36, no. 2, pp. 309–317, 2012.

[30] Z. Usman, R. I. Young, K. Case, and J. a Harding, "A manufacturing foundation ontology for product lifecycle interoperability," *Enterp. Interoperability IV Mak. Internet Futur. Futur. Enterp.*, pp. 147–155, 2010.

[31] K.-H. Lee, Y.-S. Kang, and Y.-H. Lee, "Development of Manufacturing Ontology-based Quality Prediction Framework and System: Injection Molding Process," *IE interfaces*, vol. 25, no. 1, pp. 40–51, 2012.

[32] H. K. Lin and J. a. Harding, "A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration," *Comput. Ind.*, vol. 58, no. 5, pp. 428–437, 2007.

[33] F. B. Vernadat, "Technical, semantic and organizational issues of enterprise interoperability and networking," *Annu. Rev. Control*, vol. 34, no. PART 1, pp. 139–144, 2010.

[34] H. Panetto, R. Jardim-Goncalves, and A. Molina, "Enterprise integration and networking: Theory and practice," *Annu. Rev. Control*, vol. 36, no. 2, pp. 284–290, 2012.

[35] F. Ameri, C. Urbanovsky, and C. McArthur, "A Systematic Approach to Developing Ontologies for Manufacturing Service Modeling," *Proc. Work. Ontol. Semant. Web Manuf.*, pp. 1–14, 2012.

[36] H. Panetto, M. Dassisti, and a. Tursi, "ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment," *Adv. Eng. Informatics*, vol. 26, no. 2, pp. 334–348, 2012.

[37] R. Jardim-Goncalves, J. Sarraipa, C. Agostinho, and H. Panetto, "Knowledge framework for intelligent manufacturing systems," *J. Intell. Manuf.*, vol. 22, no. 5, pp. 725–735, 2011.

[38] R. Jardim-Goncalves, C. Coutinho, A. Cretan, C. F. da Silva, and P. Ghodous, "Collaborative negotiation for ontology-driven enterprise businesses," *Comput. Ind.*, vol. 65, no. 9, pp. 1232–1241, 2014.

[39] N. Chungoora, R. I. Young, G. Gunendran, C. Palmer, Z. Usman, N. a. Anjum, A.-F. Cutting-Decelle, J. a. Harding, and K. Case, "A model-driven ontology approach for manufacturing system interoperability and knowledge sharing," *Comput. Ind.*, vol. 64, no. 4, pp. 392–401, May 2013.

[40] M. Imran and B. Young, "The application of common logic based formal ontologies to assembly knowledge sharing," *J. Intell. Manuf.*, vol. 26, no. 1, pp. 139–158, Apr. 2013.

[41] D. Kiritsis, "Closed-loop PLM for intelligent products in the era of the Internet of things," *Comput. Des.*, vol. 43, no. 5, pp. 479–501, May 2011.

[42] A. Matsokis and D. Kiritsis, "Ontology applications in PLM," *Int. J. Prod. Lifecycle Manag.*, vol. 5, no. 1, pp. 84–97, 2011.

[43] C. Legat, C. Seitz, S. Lamparter, and S. Feldmann, "Semantics to the Shop Floor: Towards Ontology Modularization and Reuse in the Automation Domain," in *Proceedings of the 19th IFAC World Congress*, 2014, pp. 3444–3449.

[44] L. F. Lin, W. Y. Zhang, Y. C. Lou, C. Y. Chu, and M. Cai, "Developing manufacturing ontologies for knowledge reuse in distributed manufacturing environment," *Int. J. Prod. Res.*, vol. 49, no. 2, pp. 343–359, Jan. 2011.

[45] W. Long, "Research on Development Method of MES Based on Component and Driven by Ontology," *J. Softw.*, vol. 5, no. 11, pp. 1228–1235, Nov. 2010.

[46] W. J. Verhagen and R. Curran, "Ontological modelling of the aerospace composite manufacturing domain," in *Improving Complex Systems Today*, Springer London, 2011, pp. 215–222.

[47] A. Ferrándiz-Colmeiro, V. Gilart-Iglesias, and F. Maciá-Pérez, "Semantic processes modelling independent of manufacturing infrastructures," in *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2010*, 2010.

[48] M. C. Suárez-Figueroa, A. Gómez-Pérez, and B. Villazón-Terrazas, "How to write and use the ontology requirements specification document," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5871 LNCS, no. PART 2, pp. 966–982, 2009.

[49] J. Heflin, "OWL Web Ontology Language Use Cases and Requirements," *W3C Proposed Recommendation*, 2003. .

[50] J. R. G. Pulido, M. a. G. Ruiz, R. Herrera, E. Cabello, S. Legrand, and D. Elliman, "Ontology languages for the semantic web: A never completely updated review," *Knowledge-Based Syst.*, vol. 19, no. 7, pp. 489–497, Nov. 2006.

[51] S. Heymans, L. Ma, D. Anicic, Z. Ma, N. Steinmetz, Y. Pan, J. Mei, A. Fokoue, A. Kalyanpur, A. Kerschenbaum, E. Schonberg, K. Srinivas, C. Feier, G. Hench, B. Wetzstein, and U. Keller, "ONTOLOGY REASONING WITH LARGE DATA REPOSITORY," in *Ontology Management*, Springer US, 2008, pp. 89–128.

[52] National Center for Manufacturing Sciences, "NCMS Collaborative Manufacturing Agenda. NCMS Document 0040RE96," 1996.

[53] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, "Ontology Based Context Modeling and Reasoning using OWL," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communication Workshops (PERCOMW'04)*, 2004.

[54] H. Zhang, Y.-F. Li, and H. B. K. Tan, "Measuring design complexity of semantic web ontologies," *J. Syst. Softw.*, vol. 83, no. 5, pp. 803–814, 2010.

[55] E. Negri, L. Fumagalli, M. Garetti, and L. Tanca, "A review of semantic languages for the conceptual modelling of the manufacturing domain," in *Proceedings of the XIX Summerschool of Industrial Mechanical Plants "Francesco Turco", Ancona, Italy, 9th-12th September 2014*, 2014.

[56] S. Brockmans, R. Volz, A. Eberhart, and L. Peter, "Visual Modeling of OWL DL Ontologies Using," pp. 198–213, 2004.

[57] O. Corcho, M. Fernández-López, and A. Gómez-Pérez, "Methodologies, tools and languages for building ontologies. Where is their meeting point?," *Data Knowl. Eng.*, vol. 46, no. 1, pp. 41–64, Jul. 2003.

[58] J. De Bruijn, L. Rubén, A. Polleres, and D. Fensel, "OWL DL vs. OWL flight: conceptual modeling and reasoning for the semantic Web," *Proc. 14th Int. Conf. World Wide Web*, pp. 623–632, 2005.

[59] S. Brockmans, R. M. Colomb, P. Haase, E. F. Kendall, E. K. Wallace, C. Welty, and G. T. Xie, "A Model Driven Approach for Building OWL DL and OWL Full Ontologies," pp. 187–200, 2006.

[60] D. Djurić, D. Gašević, V. Devedžić, and V. Damjanović, "A UML Profile for OWL Ontologies," in *MDAFA 2003/2004, LNCS 3599*, 2005, pp. 204–219.

[61] D. Gasevid, D. Djuric, V. Devediid, and V. Damjanovid, "From UML to Ready-To-Use OWL Ontologies," no. June, 2004.

[62] R. Grønmo, M. C. Jaeger, and H. Hoff, "Transformations Between UML and OWL-S," pp. 269–283, 2005.

[63] I. Kim and K. Lee, "A Model-Driven Approach for Describing Semantic Web Services: From UML to OWL-S," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 39, no. 6, pp. 637–646, Nov. 2009.

[64] J. Z. Pan, "A Flexible Ontology Reasoning Architecture for the Semantic Web," vol. 19, no. 2, pp. 246–260, 2007.

[65] C. Schlenoff, R. Ivester, and A. Knutilla, "a Robust Process Ontology for Manufacturing Systems," *Proc. 2nd Int. Conf. Eng. Des. Autom.*, pp. 7–14, 1998.

[66] B. Hardy, G. Apic, P. Carthew, D. Clark, D. Cook, I. Dix, S. Escher, J. Hastings, D. J. Heard, N. Jeliazkova, P. Judson, S. Matis-Mitchell, D. Mitic, G. Myatt, I. Shah, O. Spjuth, O. Tcheremenskaia, L. Toldo, D. Watson, A. White, and C. Yang, "Toxicology ontology perspectives.," *ALTEX*, vol. 29, no. 2, pp. 139–56, Jan. 2012.

[67] A. Uszok, J. M. Bradshaw, J. Lott, M. Johnson, M. Breedy, M. Vignati, K. Whittaker, K. Jakubowski, J. Bowcock, and D. Apgard, "Toward a flexible ontology-based policy approach for network operations using the KAoS framework," in *MILITARY COMMUNICATIONS CONFERENCE, 2011-MI*, 2011, pp. 1–8.

[68] D. Randall, "The essential features of an ontology for cyberwarfare," in *Conflict and cooperation in cyberspace*, 2013, pp. 35–48.

[69] T. Lawson, "Ontology and the study of social reality: emergence, organisation, community, power, social relations, corporations, artefacts and money," *Cambridge J. Econ.*, vol. 36, no. 2, pp. 345–385, 2012.

[70] D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus, "Querying RDF streams with C-SPARQL," *ACM SIGMOD Rec.*, vol. 39, no. 1, p. 20, 2010.

[71] J.-P. Calbimonte, H. Jeung, O. Corcho, and K. Aberer, "Enabling Query Technologies for the Semantic Sensor Web," *Int. J. Semant. Web Inf. Syst.*, vol. 8, no. 1, pp. 43–63, 2012.

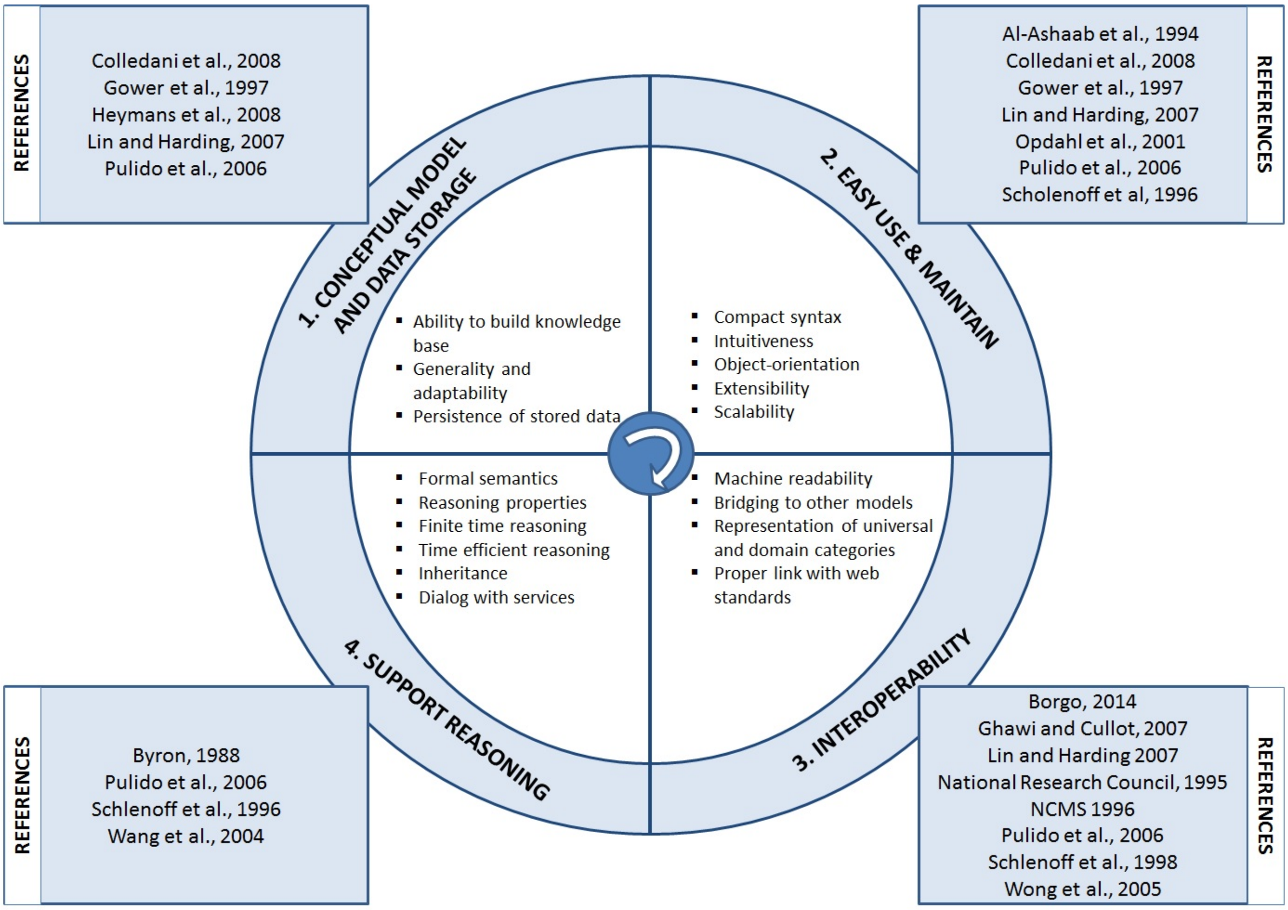
REFERENCES
Colledani et al., 2008
Gower et al., 1997
Heymans et al., 2008
Lin and Harding, 2007
Pulido et al., 2006
REFERENCES
Al-Ashaab et al., 1994
Colledani et al., 2008
Gower et al., 1997
Lin and Harding, 2007
Opdahl et al., 2001
Pulido et al., 2006
Scholenoff et al, 1996
1. CONCEPTUAL MODEL AND DATA STORAGE
2. EASY USE & MAINTAIN
Ability to build knowledge base
Generality and adaptability
Persistence of stored data
Compact syntax
Intuitiveness
Object-orientation
Extensibility
Scalability
Formal semantics
Reasoning properties
Finite time reasoning
Time efficient reasoning
Inheritance
Dialog with services
Machine readability
Bridging to other models
Representation of universal and domain categories
Proper link with web standards
4. SUPPORT REASONING
3. INTEROPERABILITY
REFERENCES
Byron, 1988
Pulido et al., 2006
Schlenoff et al., 1996
Wang et al., 2004
REFERENCES
Borgo, 2014
Ghawi and Cullot, 2007
Lin and Harding 2007
National Research Council, 1995
NCMS 1996
Pulido et al., 2006
Schlenoff et al., 1998
Wong et al., 2005

Figure 1

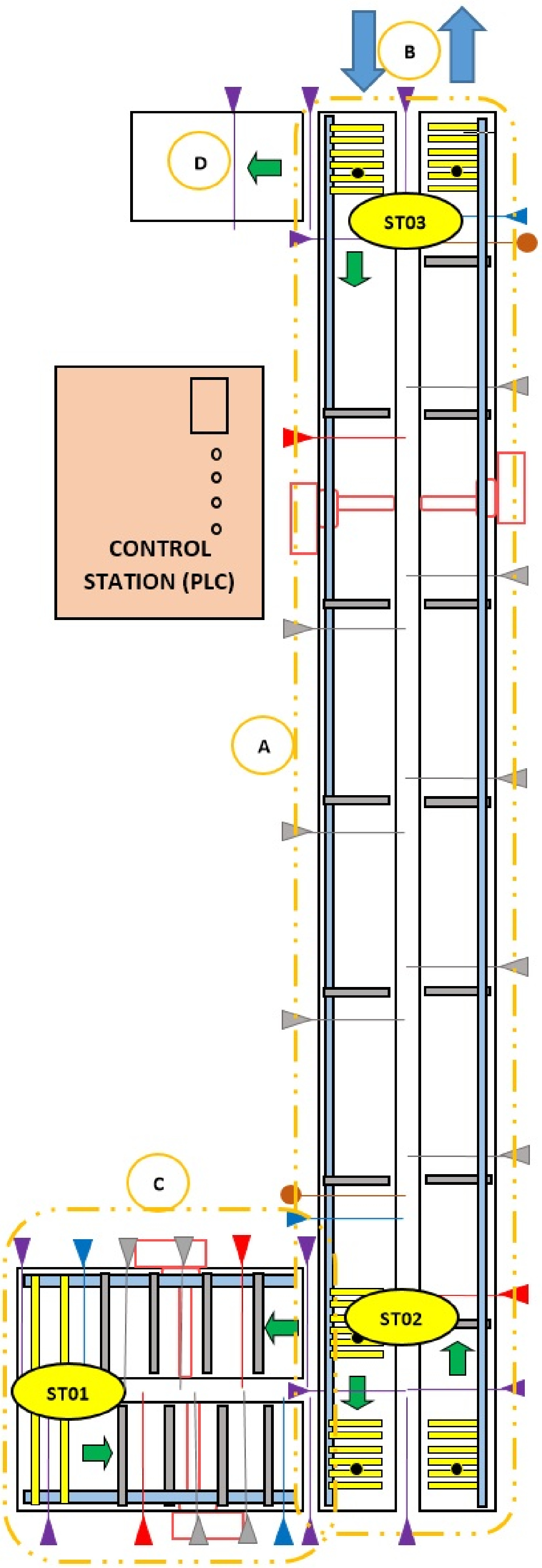

Figure 2

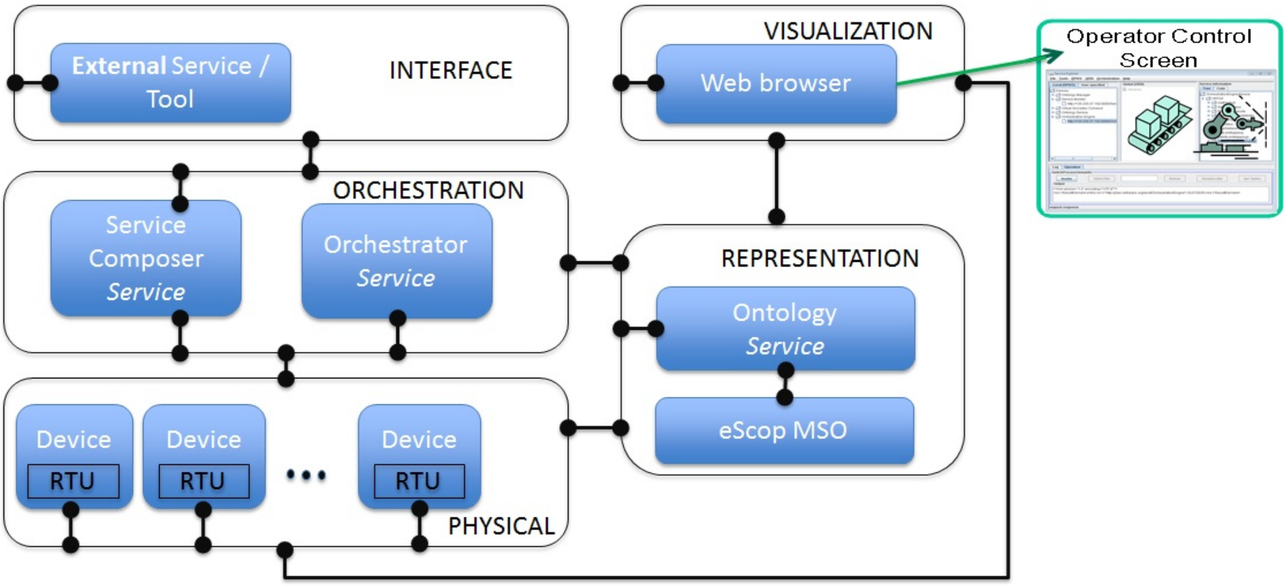

Figure 3

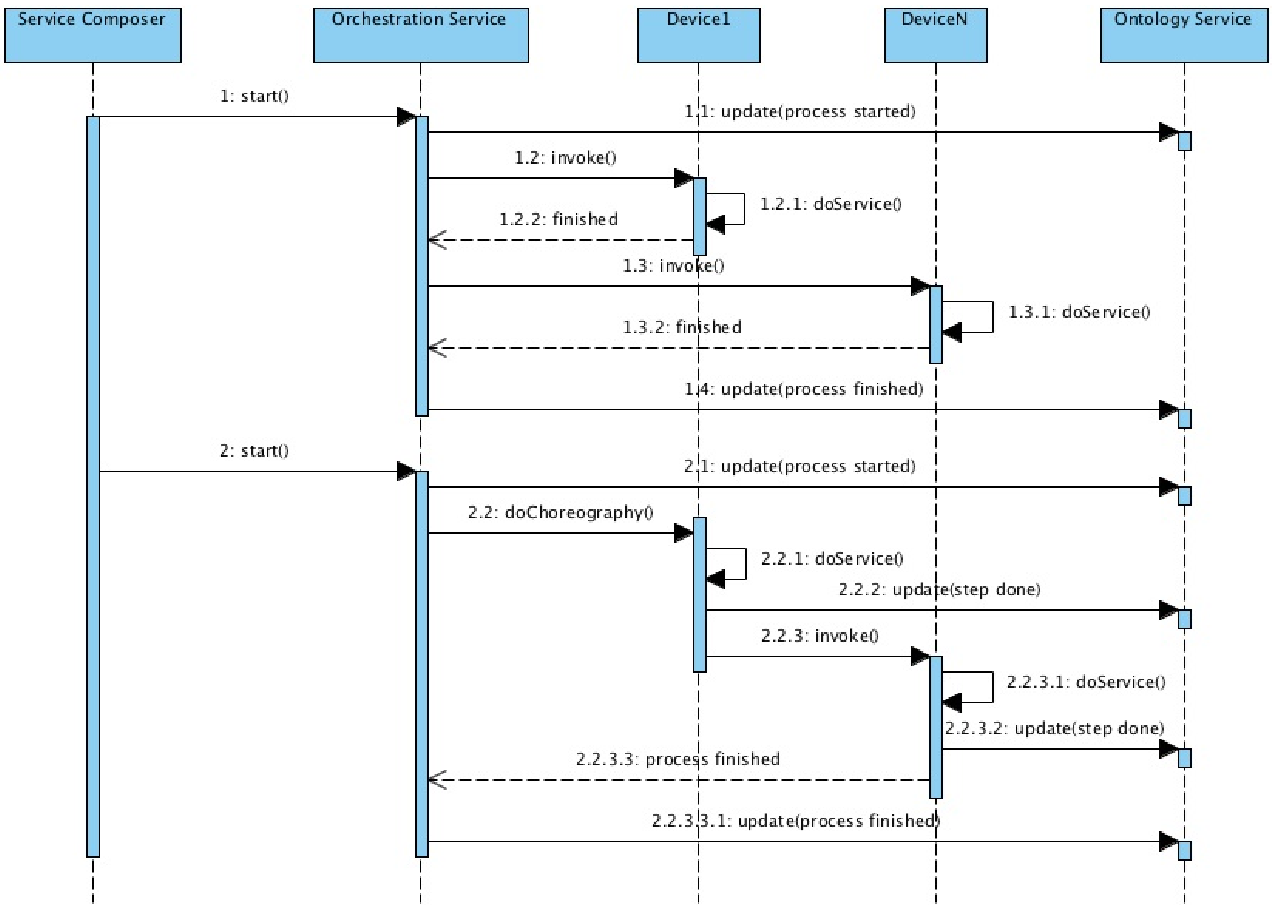

Figure 4

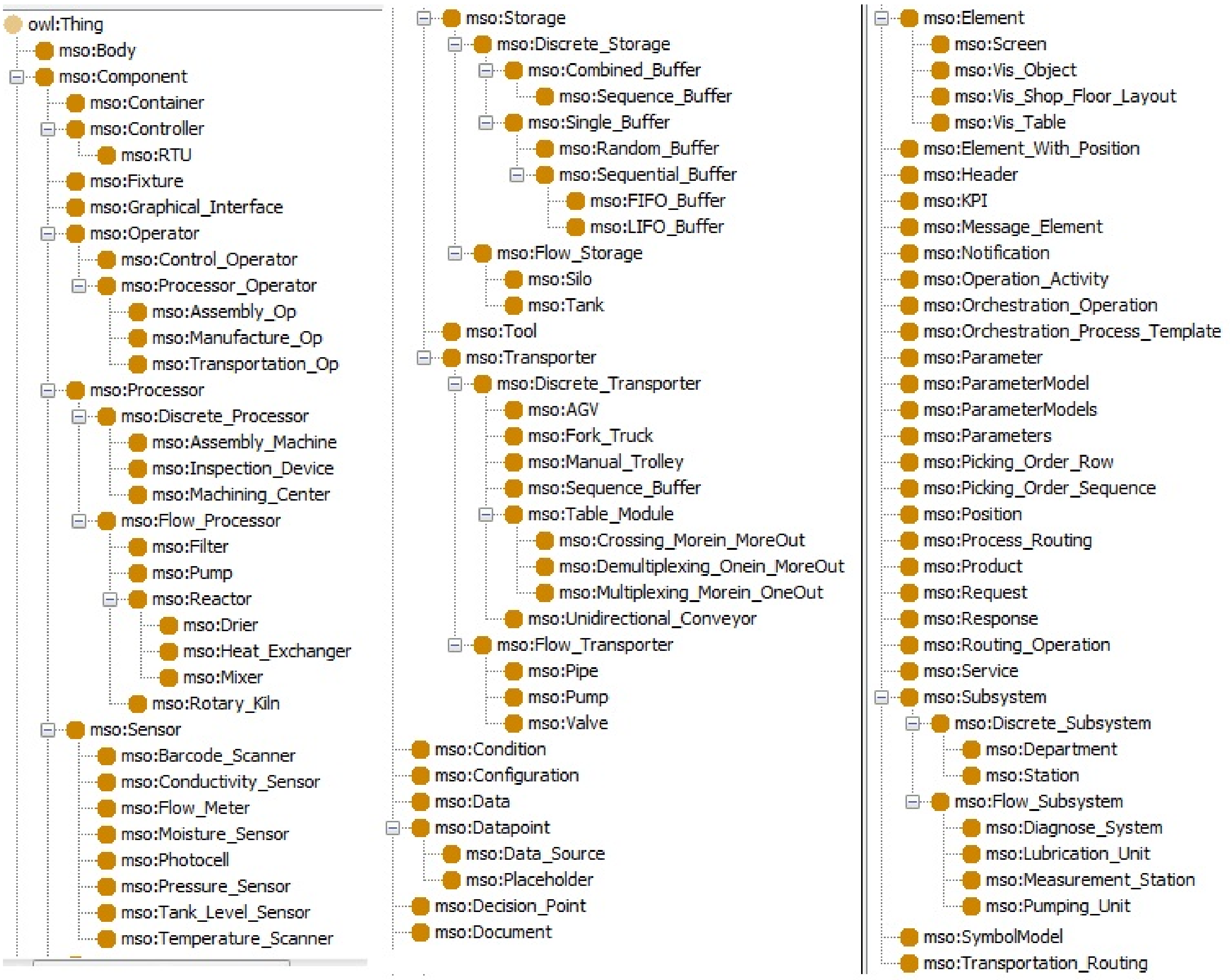

Figure 5

Table 1 - Semantic languages and features

Language	Features
KIF	Language based on first order logic. KIF is a formal language developed for the interchange of knowledge among disparate computer programs (written by different programmers, at different times, in different languages, and so forth). KIF provides the level of rigor necessary to unambiguously define concepts in the ontology, a necessary characteristic to exchange manufacturing process information. It has declarative semantics, logical comprehension, meta-knowledge, translatability, readability. It allows a conceptualization of the world in terms of objects, functions and relations.
	Web Language: No **Reasoning Support**: No
OntoLingua	Built on KIF, it allows a representation of concepts, taxonomies of concepts, n-ary relations, functions, axioms, instances and procedures.
	Web Language: No **Reasoning Support**: No
OCML - Operational Conceptual Modelling Language	Traditional syntax ontology language. Similar to OntoLingua with additional components: deductive and production rules, and operational definitions for functions. Built for developing executable ontologies and models in problem solving methods.
	Web Language: No **Reasoning Support**: Very limited
FLogic - FrameLogic	FrameLogic combines frames and first order logic, allowing to represent concepts, concept taxonomies, binary relations, functions, instances, axioms, deductive rules, objects, inheritance, polymorphic types, query methods and encapsulation.
	Web Language: No

	Reasoning Support: Constraint checking and deducting new information through structural and behavioral inheritance
Loom	Initially, it was not meant for implementing ontologies, but for general Knowledge Bases. Loom is based on Description Logics and production rules, and provides automatic classifications of concepts. Represents concepts, concept taxonomies, n-ary relations, functions, axioms and production rules.
	Web Language: No **Reasoning Support**: Very limited
DublinCore	Simplicity is both its strength and weakness, based on Logic Programming, it can be found at http://dublincore.org .
	Web Language: Yes **Reasoning Support**: Yes
SHOE - Simple HTML Ontology Extension	Extension of HTML to introduce ontologies into HTML documents, used to add semantics to web pages. It consists of object-oriented tags to provide structure for knowledge acquisition: it associates meaning with content by committing web pages to existing ontologies. It represents concepts, their taxonomies, n-ary relations, instances and deduction rules, which are used by its inference engine to obtain new knowledge. Can be found at www.cs.umd.edu/projects/plus/SHOE/ontologies.html .
	Web Language: Yes **Reasoning Support**: has inference rules
XML (S)	XML replaced SHOE. After XML, all ontology languages are built on XML. It uses first language to separate the markup of web content from web presentation. Issues related to this language: (i) it lacks semantics: designed to describe the structure of a document not the content; (ii) "is-a" relationship does not exist; (iii) attributes are not local to an object but global to a document; (iv) no notion of inheritance; (v) it defines an order in which tags appear in a document, order does not matter in an ontology; (v) difficult for machines when new vocabulary is used: no difference between polysemous terms and no possibility to combine synonymous terms.
	Web Language: Yes **Reasoning Support**: No
XOL	Very restricted language (concepts, concept taxonomies and binary relations).
	Web Language: Yes; **Reasoning Support**: No
RDF (S)	RDF provides a simple data model and the RDF schema defines a simple ontology language with classes, sub-classes, properties, sub-properties, and domain and range restrictions in RDF for expressing metadata. It is used to describe info about Web resources, to make info machine processable and to provide automated processing of Web info by intelligent agents. RDF is not very expressive (it only represents concepts, concept taxonomies and binary relations). It was created by the World Wide Web to provide meaning to data. It can be linked to any Web resource: interoperability between applications that exchange machine-understandable information on the web (interoperability = important advantage from XML, thanks to the semantics, because XML did not represent meaning). It consists of independent objects that form object-attribute-value triples (representable with a directed graph data model with nodes and edges: nodes are subject and object while the edge is a predicate), where subjects, objects and predicates are identified by URIs (even if objects may also be literals). RDFS has been introduced as a layer on top of RDF as a set of ontological modeling primitives (classes and subclasses of resources, properties and relations): this allows to set a particular vocabulary for RDF data. With the structure of classes and subclasses it allows users to publish ontologies on the Web. However, the RDF Schema is not explicit (formal) enough and still does not provide exact semantics when it comes to representing complex constraints.
	Web Language: Yes **Reasoning Support**: Some inference engines mainly for constraint checking
OIL - Ontology Inference Layer	It is an extension of RDF(S), therefore has a well-defined syntax in XML. It provides a standardized syntax for writing ontologies and a standard set of modeling primitives. Its formal semantics is based on Description Logics. It allows the automatic classification of concepts and the representation of taxonomies, binary relations, functions and instances.
	Web Language: Yes **Reasoning Support**: Yes
DAML - DARPA Agent Markup Language	It is markup language for semantic web with expressive power and a well-defined semantics for reasoning. It includes mappings to other semantic languages: SHOE, OIL, KIF, XML, RDF. It is made of two portions: ontology language (DAML + OIL) and a language for expressing constraints and inference rules (DAML-L).
	Web Language: Yes **Reasoning Support:** Yes
DAML+OIL	It is an extension of RDF(S) and has a formal semantics based on Description Logics. It allows representing concepts, taxonomies, binary relations, functions and instances. It is unambiguously computer interpretable, it supports agent interoperability and automated reasoning.
	Web Language: Yes

	Reasoning Support: Many efforts still done for reasoning
DAML-L	Logical language with a well-defined semantics and the ability to express a compact representation of constraints and rules for reasoning. **Web Language:** Yes **Reasoning Support**: It has reasoning rules
OWL	It is a defacto standard ontology language, it is compatible with SHOE and DAML+OIL and is an extension of RDF(S), but has more power to express semantics. It includes classes and operations on classes such as conjunction and disjunction and existentially and universally quantifiable variables. One of the significant features of the OWL language is its ability to make equality claims; in fact, OWL introduces constructions to state equality between classes (owl:sameClassAs) and between properties (owl:samePropertiesAs). This enables mapping between different individual ontologies: in fact OWL provides built-in ontology mapping support, that is, a particular class or property in one ontology is the same as a class or property in another ontology (owl:sameClassAs, owl:samePropertyAs): the individuals therefore have the same "identity". It is characterized by logical inference and can derive knowledge. It has more powerful reasoning from RDF: RDF has only a propositional reasoning, OWL reasoning can be about whole documents. Drawbacks: (i) it has a very complex, not efficient reasoning; (ii) it is not easy to use, (iii) it is not intuitive, (iv) it does not have built-in primitives for the (very important) part-whole relations, (v) it cannot deal with the fact that the meaning of certain words is context dependent. For this reason it comes in many flavors, of which the main ones are three: OWL FULL, OWL DL, OWL LITE, the selection criterion is to take the best for the system requirements. It can be found at www.w3.org/TR/2003/WD-owl-ref-20030331. **Web Language:** Yes **Reasoning Support**: It allows complex reasoning about documents
OWL Lite	Trade expressivity for efficiency (and guaranteed termination) of reasoning. It uses only 35 out of 40 OWL constructs and only 11 out of 33 RDF(S) constructs. In addition, some of them can only be used with limitations. **Web Language:** Yes **Reasoning Support**: Guaranteed termination and efficiency in reasoning
OWL DL	It is the W3C standard ontology language: it is the most important OWL because it is a variant of Description Logics and the reasoning may exploit many state-of-art Description Logics reasoners. It uses all 40 OWL constructs and only 11 out of 33 RDF(S) constructs, and some of them can be used with limitations. There is a balance between expressivity and computational completeness: even though more complex, the reasoning is still decidable. Properties are differentiated into data type properties (connect instances to literals) and object properties (connect class instances). It does not support customized datatypes. **Web Language:** Yes **Reasoning Support**: Decidable reasoning, many reasoners are built on this
OWL Full	The difference from OWL DL is that properties can be assigned to classes, a class can be represented as an individual or a property and vice versa. It uses all 40 OWL constructs and only 11 out of 33 RDF(S) constructs with no limitations. **Web Language:** Yes **Reasoning Support**: Reasoning is undecidable
C-OWL - Context - OWL	It is an extension of OWL: multiple OWL ontologies and relations between these ontologies (triples subject-relation-object between 2 concepts, 2 instances or 2 properties in 2 different ontologies). It consists of an ontology and the set of bridge rules where the subject concept belongs to the ontology itself. It allows any of the OWL sub-languages but the 2 languages in the mapping should be the same sub-language (no interoperability between the different flavors of OWL is supported). **Web Language:** Yes **Reasoning Support**: Yes
OWL-Eu	Extension of the OWL DL to support customized data types by extending OWL data ranges with unary data type expressions. **Web Language:** Yes **Reasoning Support:** Yes
OWL-E	Extension of the OWL DL to support customized data type predicates with a n-ary extension of the OWL data range. **Web Language:** Yes **Reasoning Support:** Yes
OWL Flight	It is loosely based on OWL, but the semantics is grounded on Logic Programming and not on Description Logics and borrows the constraint-based modeling style common in databases. It does not have: enumerated classes, individual (in)equality assertions, complements, property restrictions. It adds: unique name assumption, cardinality constraints, property value constraints, more elaborate treatment of datatypes (following OWL-E).

	Web Language: Yes **Reasoning Support**: Reasoning is often undecidable
UML	UML is the standard for Object modeling as unique modeling language; UML profiles may be used to represent the characteristics of the OWL, because it helps to express, communicate, validate the design and development of the ontology. It is theoretically possible to transform it into OWL (ontology automatically imported as class diagram) through an XMI file (XML Metadata Interchange) via an Extensible Stylesheet Language Transformation sheet (XSLT).
	Web Language: Yes **Reasoning Support**: No