Survey on Ontology Languages

Diana Kalibatiene and Olegas Vasilecas

Vilnius Gediminas Technical University, Sauletekio al. 11, LT-10223
{diana.kalibatiene,olegas.vasilecas}@vgtu.lt

Abstract. Nowadays a number of papers are presented on the research for the ontology application for a business system modelling. For this purpose formal and executable ontologies earn a lot of attention. However, formality and executability of an ontology depends on a language, which is used to present it. This paper presents a never completely account of languages that have been used for the research community for representing ontologies. The most popular four ontology languages (KIF, OWL, RDF + RDF(S) and DAML+OIL) are reviewed. Their advantages and disadvantages are discussed. Finally, thirteen comparison criteria are distinguished and chosen ontology languages are compared. The discussion is also presented in the paper.

Keywords: Ontology, ontology language, comparison, KIF, OWL, RDF + RDF(S), DAML+OIL.

1 Introduction

In recent years, several languages have been developed to represent knowledge by ontologies. The construction of these (ontology) languages is evolving according to a layered approach, since in this layer reasoning and inference are laid. These languages must meet a number of requirements to be useful for business system modelling. According to [1], "*it must have*:

1. *a reasonably compact syntax;*
2. *a well defined semantics so that one can say precisely what is being represented;*
3. *sufficient expressive power to represent human knowledge;*
4. *an efficient, powerful, and understandable reasoning mechanism;*
5. *must be usable to build large knowledge bases."*

Moreover, an ontology language "*must describe meaning in a machine-readable way. Therefore, an ontology language needs not only to include the ability to specify vocabulary, but also the means to formally define it in such a way that it will work for automated reasoning*" [2].

Nowadays, a big emphasis is placed on web-based ontology languages. Since the web is decentralised, the languages must allow defining diverse vocabularies and letting them evolve [2].

In this paper, we discuss the most popular four ontology languages: KIF, OWL, RDF + RDF(S) and DAML+OIL. We present criteria for a comparison of ontology languages and discuss their strengths and weaknesses in relation to the chosen criteria.

J. Grabis and M. Kirikova (Eds.): BIR 2011, LNBIP 90, pp. 124–141, 2011.

The last paper is structured as follows. Section 2 presents concept of ontology. Section 3 overviews ontology languages. Section 4 presents related work on the comparison of ontology languages. Section 5 presents selecting criteria for comparison of ontology languages. And Section 6 presents the discussion on the survey obtained.

2 Concept of Ontology

Nowadays, there are a lot of definitions of ontology. It is mostly depends on the task how and why ontology is used. Classical definition, borrowed from philosophy, says that ontology means a systematic account of Existence.

According to N. Guarino [3], "*an ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i. e. its ontological commitment to a particular conceptualisation of the world*". Based on such definition, an ontology consists of concepts, their definitions, relationships between concepts and constrains expressed as axioms.

However, the modern definition of ontology is extended in term of instances (in Protégé[1]-Frames ontologies) or individuals (in Ontology Web Language (OWL) [4] based ontologies), which represent objects in the domain of discourse.

According to the components, ontology defines the basic concepts, their definitions and relationships comprising the vocabulary of an application domain and the axioms for constraining interpretation of concepts and expressing complex relationships between concepts [5], [6]. Some authors, like in [7], distinguish properties from concepts also.

According to the content of a business domain knowledge, ontologies can be: *lightweight*, which describes a hierarchy of concepts related by particular relationships (e.g., is-a, part-of, etc.); *light heavyweight*, in which constraints are added to restrict the values of concepts and relationships, like cardinality constraints, possible length, etc.; and *heavyweight*, in which suitable axioms are added in order to express and restrict complex relationships between concepts and to constrain their intended interpretation.

The most existing ontologies, like WordNet[2], which can be used as a lexical ontology, Protégé ontologies (not all), ontologies presented in [8] and [9], DBpedia[3], are lightweight or light heavyweight, since those have no axioms. In heavyweight ontologies axioms defined in a framework of a description logics [10], in some kind of logic language, like KIF[4] in Protégé[1]–Frames ontology [11] and SUMO[5]. However, light heavyweight and heavyweight ontologies are the most useful for business systems modelling, since they are restricted and contain axioms and constraints, which can be used for business rules and business constraints modelling [5], [6].

[1] `http://protege.stanford.edu`
[2] `http://wordnet.princeton.edu/`
[3] `http://dbpedia.org/About`
[4] `http://logic.stanford.edu/kif/kif.html`
[5] `http://www.ontologyportal.org/`

Authors of this paper suggest using the definition of ontology proposed in [12] to define main criteria for ontology languages. His definition is as follows:

"***An ontology is a formal, explicit specification of a shared conceptualization***."

In this context, *conceptualization* refers to an abstract model of some phenomenon in the world, like business, that identifies that phenomenon's relevant concepts, i.e. business concepts. *Explicit* means that the type of concepts used and the constraints on their use are explicitly defined and *formal* means that the ontology should be machine understandable. Different degrees of formality are possible. Large ontologies such as WordNet[6] provide a thesaurus for over 100,000 terms explained in natural language. On the other end of the spectrum is CYC[7], which provides formal axiomating theories for many aspects of commonsense knowledge. *Shared* reflects the notion that an ontology captures consensual knowledge – it is not restricted to some individual but is accepted by a group.

3 Ontology Languages

In computer science, *ontology languages* are formal languages used to construct ontologies. They allow the encoding of knowledge about specific domains and often include reasoning rules that support the processing of that knowledge. Ontology languages are usually *declarative languages*, are almost always generalizations of *frame languages*, and are commonly based on either *first-order logic* or on *description logic* [13]. A part of existing languages are implemented into ontology development and management tools (ontology tools) or used in particular methodologies. The most interesting thing in ontology engineering is that despite the existence of many methodologies, tools and languages, it is not easy to select an appropriate ontology development technique. Sets of ontology languages, ontology tools and developing methodologies are presented in Fig. 1. You may notice that not all available methodologies, languages and tools can be selected for ontology development. This situation exists because some ontology development methodologies were built for specific ontology languages, some ontology development methodologies could be used only with certain tools. As a result, options of selecting an appropriate ontology development technique are restricted, and only certain approaches can be used for ontology development. Authors of [14] and [15] propose facilitating ontology development by constant evaluation of steps in the process of ontology development and the user guide throughout the process to solve the above problem.

According to [16], ontology languages can be divided into two categories. The first category is ***traditional ontology languages***. They are: languages based on *first-order predicate logic* (KIF, CycL), *frame-based languages* (Ontolingua, F-logic and OCML), *description logic (DL) based languages* (Loom), and other languages. The second category is Web standards, which are used to facilitate interchange on the Internet, and ontology languages, which are web standards compatible, are named ***Web-based ontology languages***. However, particular languages can be assigned to

[6] `www.cogsci.princeton.edu/~wn`

[7] `www.cyc.com`

both categories, i.e. they are based on a particular Web standard and a particular paradigm, like frames or first-order predicate logic. For example, OWL DL is based on RDF and description logic. It is presented in Fig. 2. We are going to pay more attention to the languages, which belong to both sides.

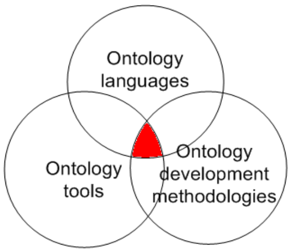

Fig. 1. Ontology development methodologies, languages and tools

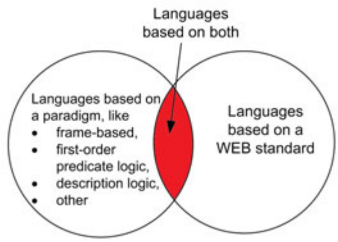

Fig. 2. Ontology languages

Below we present several ontology languages, which are going to be compared in the next section of this paper. Some advantages and disadvantages and an example are going to be presented for each chosen language.

3.1 KIF (SUO-KIF SUMO Language)

Knowledge Interchange Format (KIF)[8] is a computer-oriented language for the interchange of knowledge among disparate programs. The basis for the semantics of KIF is a *conceptualization* of the world in terms of objects and relations among those objects. Its features are as follows [17]:

- *declarative semantics* – possibility to understand the meaning of expressions without appeal to an interpreter for manipulating those expressions;
- *logically comprehensive* – the expression of arbitrary sentences in the first-order predicate calculus;
- *meta-knowledge* – the representation of knowledge about the representation of knowledge;
- the representation of *nonmonotonic reasoning rules*;
- *readability* – the definition of objects, functions, and relations;
- *implementability* – although KIF is not intended for use within programs as a representation or communication language, it should be usable for that purpose if so desired.

It was originally created in the DARPA knowledge Sharing Effort [18]. There have been a number of versions of KIF and several languages based on KIF, like SUO-KIF[9], are created. Table 1 presents an example of an ontology axiom defined in SUO-KIF: "*A rail vehicle is a vehicle designed to move on railways.*"

[8] http://logic.stanford.edu/kif/
[9] http://suo.ieee.org/SUO/KIF/suo-kif.html

Table 1. An example of SUMO axiom defined in SUO-KIF[10]

```
(subclass RailVehicle LandVehicle)
   (documentation RailVehicle
        "A Vehicle designed to move on &%Railways.")
   (=>  (instance ?X RailVehicle)
        (hasPurpose ?X
         (exists (?EV ?SURF)
           (and  (instance ?RAIL Railway)
                 (instance ?EV Transportation)
                 (holdsDuring (WhenFn ?EV)
                         (meetsSpatially ?X ?RAIL))))))
```

The main disadvantages of KIF are: a *high expressiveness* of KIF complicates the building of fully conforming systems; the resulting systems tend to be "*heavyweight*", i.e. they are larger and in some cases less efficient than systems that employ more restricted languages[7].

3.2 OWL (Lite, DL, Full)

OWL is a standard ontology language for the semantic web. It is a language extension of RDF Schema. OWL is compatible with early ontology languages, including SHOE, DAML+OIL, and provides the engineer more power to express semantics. It includes conjunction, disjunction, existentially, and universally quantified variables, which can be used to carry out logical inferences and derive knowledge. An example of an OWL is presented in Table 2. However, OWL has the following drawbacks [19]:

1. Some constructs are very *complex*, therefore three sublanguages is designed [4]:

– OWL Lite – supports a classification hierarchy and simple constraint features, like cardinality constraints. The advantage of this is that is both easier to grasp (for users) and easier to implement (for tool builders) [20]. The disadvantage is a restricted expressivity.
– OWL DL – includes all OWL language constructs with restrictions such as type separation (a class cannot also be an individual or property). It corresponds with description logics. The advantage is that it permits efficient reasoning support [20]. However, full compatibility with RDF is lost.
– OWL Full – supports maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. OWL Full fully compatible with RDF, both syntactically and semantically [20]. However, it becomes undecidable and not suitable for complete reasoning in particular cases.

2. Reasoning is not efficient as there is a trade-off against time-complex cost.

– It is *not intuitive*, need to be owl-savvy to build efficient knowledge constructions.

---

[10] http://www.obitko.com/tutorials/ontologies-semantic-web/knowledge-interchange- format.html

Table 2. An example of Pizza ontology – OWL class [21]

```
    <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
            <owl:Class rdf:about="#MozzarellaTopping"/>
            <owl:Class
           rdf:about="#PeperoniSausageTopping"/>
            <owl:Class rdf:about="#JalapenoPepperTopping"/>
            <owl:Class rdf:about="#TomatoTopping"/>
            <owl:Class rdf:about="#HotGreenPepperTopping"/>
        </owl:unionOf>
    </owl:Class>
```

3.3 RDF

The *Resource Description Framework* (RDF)[11] is a standard for describing resources on the web developed by the World Wide Web Consortium (W3C). It is designed to be read and understood by computers not displayed to people. It is suitable for describing any web resource and as such it provides interoperability between applications that exchange machine-understandable information on the web. It is written in Extensible Markup Language (XML). RDF is becoming a widely recognized language and a representation formalism that can serve as a worldwide interlingua for information interchange.

RDF is based on the idea of identifying things on the Web via URI references and expressing statements about resources in terms of simple properties and property values [22]. Each RDF statement consists of a triplet: a *subject*, a *predicate* (also called a property) and an *object*, or, as stated in some references like [2], an object, an attribute and a value. Its goal is to add formal semantics to the web and provide a data model and syntax convention for representing the semantics of the data in a standardized manner. The relationships among resources are described in terms of the named properties and values. An example of RDF is presented in Table 3.

Table 3. An example of a describing a record shop in RDF[12]

```
    <rdf:Description
    rdf:about="http://www.recshop.fake/cd/Hide your heart">
        <cd:artist>Bonnie Tyler</cd:artist>
        <cd:country>UK</cd:country>
        <cd:company>CBS Records</cd:company>
        <cd:price>9.90</cd:price>
        <cd:year>1988</cd:year>
    </rdf:Description>
```

RDF has significant advantages over XML. They are:

- The *object-attribute structure* – all objects are independent entities.
- The *RDF model* – RDF describes an independent layer.

[11] http://www.w3.org/RDF/

[12] http://www.w3schools.com/rdf/default.asp

- An *extensible object-oriented type system* – the RDF schema (RDFS) has been introduced as a layer on the top of the basic RDF model. The RDFS can be thought of as a set of ontological modelling primitives. XML lacks this layer and some developers building a layer on the top of XML to integrate ontological primitives.
- *Developing of RDF vocabulary* – RDFS lets developers define a particular vocabulary for RDF data and specify the kind of object to which these attributes may be applied.

RDF played an important role as a basis for DARPA Agent Markup Language (DAML)[13], whose layers of logic are to be built on the top of the basic RDF framework. However, the disadvantages of RDF are: lack of affection inference mechanism and the lack of formal underlying model semantics [23].

3.4 The Ontology Interchange Language (OIL)

The *Ontology Interchange Language* (OIL) is a full-fledged Web-based ontology language based on RDFS [24]. It was created according to these requirements [25] as follows by a group of (mostly) European researchers[14] it must: a) be *highly intuitive* to the human user; b) have a *well-defined formal semantics* with established reasoning properties to ensure completeness, correctness, and efficiency; c) have a *proper link* with existing Web languages such as XML and RDF to ensure interoperability. Not all ontology languages, like CycL[15], KIF and Ontolingua[16], meet these requirements [25]. The main advantages of OIL are as follows:

- The *frame-based* and *object-oriented* modelling paradigms. OIL is based on the notion of a class and its superclasses and attributes. Relations can be defined as independent entities.
- *Well-defined formal semantics* based on DL. In addition the meaning of any expression can be described in a mathematic precise way, which enables reasoning with concept description and the automatic derivation of classification taxonomies.
- *Based on WEB standards*. OIL has a well-defined syntax in XML. It is also defined as an extension of the RDF and its extension schema (RDFS).
- It offers *different levels of complexity*. It has four layers: Core OIL, Standard OIL, Instance OIL (Standard OIL + RDFS) and Heavy OIL [25].

The following OIL expression presented in Table 4 defines *herbivore as a class, which is a subclass of animal and disjoint to all carnivores*.

However, authors of [26] and [27] find the following disadvantages of OIL. It is impossible: to define the default-value, to provide the meta-class and to support the concrete domain. Translating from OIL to RDF and back is no longer guaranteed to give an identical ontology from a modelling perspective (though semantic equivalence is still preserved).

---

13 `http://www.daml.org/`
14 `http://www.ontoknowledge.org/oil`
15 `http://www.opencyc.org/doc/`
16 `http://ksl.stanford.edu/software/ontolingua/`

Table 4. An example of a herbivore in OIL [25]

```
    <rdfs:Class rdf:ID="herbivore">
        <rdf:type
            rdf:resource="http://www.ontoknowledge.org/oil/
RDFSschema/#DefinedClass"/>
        <rdfs:subClassOf rdf:resource="#animal"/>
        <rdfs:subClassOf>
            <oil:NOT>
                <oil:hasOperand rdf:resource="#carnivore"/>
            </oil:NOT>
        </rdfs:subClassOf>
    </rdfs:Class>
```

3.5 The DARPA Agent Markup Language (DAML[9])

The *DARPA Agent Markup Language* (DAML) is a US Government-sponsored endeavour aimed to develop a language and tools to facilitate the concept of the Semantic Web. DAML consists of two portions, the ontology language and a language for expressing constraints and adding inference rules. It also includes mappings to other semantic web languages such as SHOE, OIL, KIF, XML, and RDF.

The DAML language is an extension of XML and RDF. The latest release of the language (DAML+OIL) provides a rich set of constructs with which to create machine readable and understandable ontologies and to mark-up information. The ontology language (DAML +OIL) has a well-defined model-theoretic semantics as well as an axiomatic specification that determines the language's intended interpretations. This makes it an unambiguously computer-interpretable language.

The DAML Inference Language is a logical language with a well-defined semantics and the ability to express constraints and rules for reasoning. An example of a constraint is presented in Table 5 using `daml:Restriction` clause.

Table 5. An example of a process[17] description in DAML

```
    <daml:Class rdf:about="Process">
        <rdfs:comment>
            A Process can have at most one name, but names
need not be unique.
        </rdfs:comment>
        <rdfs:subClassOf>
            <daml:Restriction daml:maxCardinality="1">
                <daml:onProperty rdf:resource="#name"/>
            </daml:Restriction>
        </rdfs:subClassOf>
    </daml:Class>
```

[17] http://www.ai.sri.com/daml/ontologies/services/1-0/Process.daml

DAML+OIL is the result of a merger between DAML and OIL. It is specifically designed for use on the Web. And it exploits XML and RDF, adding the formal rigor of description logic. It takes an object-oriented approach, describing the structure in terms of classes and properties. According to [28], from a formal point of view, DAML+OIL can be seen to be equivalent to a very expressive description logic (DL), with a DAML+OIL ontology corresponding to a DL terminology.

DAML+OIL provides mechanisms for the explicit representation of services, processes, and business models [29].

4 Related Work on the Comparison of Ontology Languages and Ontologies

There are a number of works, where ontology languages and ontologies are compared according to particular criteria. However, each survey is centred on a particular view, like structure, implementation, components, etc., according to which a comparison is made. In this section several works on the comparison of ontology language are presented. For this see Table 6.

As can be seen from Table 6, different criteria are suggested to compare ontology languages. However, the following things are the most important:

- possibility to implement the language or existence of particular tools implementing the selected language and allowing to manage a created ontology;
- optimal number of constructs allowing to define a particular domain of interest;
- possibility to transform a defined ontology into another language.

Table 6. Related work on the comparison of ontology languages

Criteria / Source, Compared languages	Main points of the comparison	Criteria
[2] KIF, F-Logic, Dublin Core, CYC, XML, RDF(S), $(KA)^2$*, SHOE, OIL, DAML, OWL,	There are not defined concrete criteria. Authors present an overview of ontology languages.	Authors present the following information about languages: Developers; A particular features; Main purpose of a language; Formality; Example; A particular advantages and disadvantages.

Table 6. (*continued*)

[16] CycL, Ontolingua, F-Logic, OCML, LOOM, Telos, RDF(S), OIL, DAML+OIL, XOL, SHOE	Propose quality evaluation framework. Two criteria for ontology language comparison are used: *conceptual basis* of the language and *external representation* of the language. However, they apply only the second criterion in their comparative study.	1. Underlying conceptual basis of the language 2. External representation of the language (domain appropriateness (expressive power and perspectives**), participant knowledge appropriateness, knowledge externalizability appropriateness, comprehensibility appropriateness (number of constructs and abstraction mechanism) and technical actor interpretation appropriateness (formal syntax, formal semantics, inference engine and constraint checking)).
[30] RDF/XML, KIF, Frame-CG and Formalized-English	Shows how RDF/XML, KIF, Frame-CG (FCG) and Formalized-English (FE) can be used for knowledge representation cases. The author highlights various inadequacies, advantages, translations of the selected languages. The article [30] is suggested for knowledge providers as a guide for knowledge representation, and for developers as a list of cases for notations and inferences engines.	1) Conjunctive existentially quantified sentences; 2) Contextualization; 3) Universal quantification; 4) Lambda abstraction, Percentage, Possibility, Valuation; 5) Negations, Exclusions and Alternatives; 6) Collections and Quantifier precedence; 7) Intervals; 8) Function Calls and Lists; 9) Higher-order statements; 10) Declarations and Definitions
[32] WSMO (Web Service Modelling Ontology) and OWL-S	According to authors of [32], WSMO presents some important advantages when compared to OWL-S: a) its conceptual model has a better separation of the requester and provider point of view, b) includes the orchestration of a Web Service enabling its static or dynamic reuse; c) provides formal semantics for the choreography of Web Services and allows multiple ways of interacting with a given service; d) provides a better language layering.	6. Separation of provider and requester point of view 7. Use of non-functional properties 8. Range of non-functional properties 9. Description of requests 10. Mediation 11. Description of orchestration 12. Maturity of the externally visible behaviour specification 13. Formal semantics for the choreography description 14. Multiple choreographies 15. Grounding *16.* Languages

Table 6. (*continued*)

[31] KAoS, Rei and Ponder	The authors suggest adopting ontologies to: a) simplify the task of governing the behaviour of complex environments; b) to simplify description of policies and facilitate the analysis and the careful reasoning over them; c) to simplify the access to policy information	1) *expressiveness* to handle the wide range of policy requirements arising in the system being managed, 2) *simplicity* to ease the policy definition tasks, 3) *enforceability* to ensure a mapping of policy specifications and implementation to various platforms, 4) *scalability* to ensure adequate performance, and 5) *analyzability* to allow reasoning about policies.

* The knowledge annotation initiative of knowledge acquisition

** Perspectives are structural (S), functional (F), behavioural (B), rule (R), Object (O), communication (C), Actor (A).

5 Selecting Criteria for Comparison of Ontology Languages

Based on the related work we use the criteria for the comparison of ontology language as follows:

- ***Constructs or conceptualisation of a domain*** – constructs constitute vocabulary of a domain. Each language uses its own constructs, i.e. types of entities and relations that are considered to exist.
- ***Specification perspective*** – different languages may focus on different perspectives, and may provide constructs for only some perspectives. Authors of [16] define seven specification perspectives: a structural (a static structure); a functional (processes, activities and transformations); behavioural (states and transitions between them); a rule (rules for certain processes, activities, entities); an object (objects, processes and classes); a communication (language actions, meaning and agreement); and actor and role (actors, roles, societies, organizations). A structural perspective is the most important in ontologies.
- ***Expressiveness*** – possibility to express semantics (domain knowledge). I.e. according to the content of a domain knowledge, which can be expressed by a language, it can be lightweight, light heavyweight, and heavyweight (see Section 2). Another important aspect is lexical support – a capability for lexical referencing of elements (e.g., synonyms).
- ***Comprehensibility*** – how easily the language can be understood by the audience [16]. Important aspects are the support of abstraction mechanisms (hiding details), uniform constructs and a reasonable number of phenomena. In [33] authors described four standard hierarchical relations: classification, aggregation, generalization and association. Aggregation is not well supported in many ontology languages [16].
- ***Formality*** – a formal language is made of three components: the syntax (e.g. rules for determining the grammatical well-formedness of sentences), the semantics (e.g. rules for interpreting sentences in a precise, meaningful way within the domain

considered), and the pragmatic (e.g. rules for explaining how to use the language and for inferring useful information from the specification). Otherwise a language is informal.

- ***Inference engine*** – possibility to inference new knowledge from the existing, i.e. possibility of reasoning.
- ***Constraint checking*** – existence of a constraint checking mechanism.
- ***Implementation*** – is a language implemented into a particular tool or not?
- ***Mapping*** with other languages. Nowadays, the feature of a language to map with other language is significant, because of reusability of knowledge and interoperability.

a – Each subsequent sublanguage has its own additional features (http://www.w3.org/TR/owl-features/)

b – Properties can be used to state relationships between individuals or from individuals to data values. Examples of properties include hasChild, hasRelative, hasSibling, and hasAge. Property hierarchies may be created by making one or more statements that a property is a subproperty of one or more other properties (see rdfs:subPropertyOf:).

c – KIF accommodates varying capabilities and/or computational constraints while providing a migration path from more restrictive to more expressive.

d – SWRL: A Semantic Web Rule Language combining OWL and RuleML http://www.daml.org/2004/04/swrl/

e – classification (CL), aggregation (AG), generalization (G), association (AS)

1. ***Paradigm used*** – as presented in Fig. 2, languages can be frame-based, based on first-order predicate logic, description logic and other languages.
2. ***Web standard used*** – some ontology languages are created on the basis of Web standards. Therefore, ontology languages can be initial or based on e.g. derived.
3. ***Standard*** – is the language accepted as a standard?
4. ***Popularity*** – we add this criterion to determine the popularity of a language. We determine it according to the number of links in Google Scholar.

As can be seen, thirteen criteria are selected to compare ontology languages. However, it is important to note that some of those criteria are dependent from other. In this paper we are not presenting a deep analysis of interdependence of criteria. However, it is a topic for future research.

1. Only formal language can be implemented. However, not all formal languages are implemented till now.
2. Formality of a language influences its expressiveness. As a rule, the more a language is formal, the less it is expressive. For example, not all formal languages allow defining of synonym or aggregation relationships.
3. Only formal languages have inference engine. In non-formal or partially formal languages explicit inference can't be warranted. For example, in OWL Full inference is not clear in partial cases. As said in [34]: "It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full."

Table 7 presents the comparison of four ontology languages according to the selected criteria. Though, the paper presents DAML and OIL, we think that it is reasonable to analyse features of DAML+OIL, since it joins both features of DAML and OIL. Table 8 presents tools implementing analysed languages.

Table 7. Comparison of ontology languages

Ontology language / Criteria	KIF	OWL Lite	OWL DL	OWL Full	RDF + RDF(S)	DAML + OIL
1. *Constructs or conceptualisation of a domain*	The three layered syntax: *basic characters* (BC); *lexemes* (L) combined from BC; and *expressions* combined from the L.	Main features[a]: a *class* defines a group of individuals with the same *properties*[b]; *individuals*, instances of classes.			Triplet statements *subject – predicate – object*	Structure described in terms of *classes* and *properties*.
2. *Specification perspective*	Structural	Structural			Structural	Structural
3. *Expressiveness*	High	Weak	Medium	High	Medium	*Kinds of axiom* and *Kinds of class constructor*
4. *Comprehensibility*[e]	CL[e]: class, member, etc.; AG: composition(); G: SubClassOf(), SubSet(); AS: see KIF	CL[e]: class, subject area; AG: OWLimports: OWLOntology [0..*]; G: SubClassOf; AS: OWLhasValue, etc.			CL[e]: rdf:type, rdfs:Class; G: rdfs:subClassOf; AS: see RDF	CL[e]: rdf:type, rdfs:Class; G: rdfs:subClassOf; AS: see RDF
5. *Formality* – *Syntax*	Formal	Formal		Formal	Formal	Formal
5. *Formality* – *Semantics*	Formal	Formal		not in all cases	Formal	Formal
6. *Inference engine*	yes	yes	yes	Inference problems [35]	lack of affection inference mechanism	yes
7. *Constraint checking*	weak [c]	good	good	good	weak	supports SWRL[d]
8. *Mapping* with other languages	designed for use in the interchange of knowledge among disparate computer systems	RDF		RDF	OWL, DAML+OIL	RDF
9. *Paradigm used*	First-order predicate logic	DL		Cannot be translated to DL	Object-oriented structure	DL
10. *Web standard used*	no	RDF			XML	XML and RDF
11. *Standard*	yes	yes			yes	no
12. *Popularity*	29 100	429 000			137 000	10 500

Table 8. Implementation of ontology languages

Ontology language	**Tool**
KIF	*EPILOG* – a common lisp inference system. Epilog 2 is not available for public download. A Web Site: <http://www.cs.rochester.edu/research/epilog/>. *JKP* – a Java KIF Parser which can parse ASCII strings representing sentences in a subset of KIF into a Java representation. It is free source. A Web Site: <http://www.cs.umbc.edu/csee/research/kif/jkp/>. *Protégé*[1] – a variant of KIF, which is basis of Protégé Axiom Language (PAL), used to describe Protégé constraints. It is free source.
OWL	*OWL Validator* – a tool to check OWL mark-up for problems beyond simple syntax errors. It accepts ontologies written in RDF/XML, OWL/XML, OWL Functional Syntax, Manchester OWL Syntax, OBO Syntax, or KRSS Syntax. A Web Site: <http://www.w3.org/2001/sw/wiki/OWL_Validator>. *Protégé*[1]
RDF + RDF(S)	*Jena Toolkit* – a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine. A Web Site: <http://jena.sourceforge.net/>. *Sesame* – an open-source Java framework for storage and querying of RDF data. A Web Site: <http://www.openrdf.org/>. *KAON* – an open-source ontology management infrastructure targeted for business applications. A Web Site: <http://kaon.semanticweb.org/>. *TRIPLE* – an RDF query, inference, and transformation language for the Semantic Web. TRIPLE allows the semantics of languages on top of RDF to be defined with rules. A Web Site: <http://triple.semanticweb.org/>. *OntoEdit* – a tool which enables creating, browsing, and modifying ontologies. The Ontology Engineer is supported to transform the conceptual representation to all major ontology representation languages, like RDF(S), XML, DAML+OIL or F-Logic. It is a commercial tool, but a demo version, supporting up to 50 concepts, is available. A Web Site: <http://www.ontoprise.de/>. *Protégé*[1] – an ontology editor and a knowledge-base editor.
DAML + OIL	*OilEd* – an ontology editor allowing the user to build ontologies using DAML+OIL, it includes a FaCT (Fast Classification of Terminologies) reasoner for inferencing. *DAML Viewer* – a small Java tool for quick navigation through a DAML+OIL ontologies. A Web Site: <http://www.daml.org/viewer/>. *DAMLJessKB* – a library which uses RDF API to read in DAML files and feeds the triples into JESS, a Java-based expert system. The rules in the expert system apply the DAML semantics to the triples, creating new facts. A Web Site: <http://edge.cs.drexel.edu/assemblies/software/damljesskb/>. *SweetJess* – Semantic Web Enabling Technlogies for Jess. It Converts RULEML into DAML RuleML and then converts this into JESS Rules. A Web Site: <http://userpages.umbc.edu/~mgandh1/>.

Many of RDF tools, like Jena, Sesame, OntoEdit, support DAML+OIL.

Protégé-Frames ontologies can be exported into a variety of formats including RDF(S), OWL, and XML Schema.

6 Discussion

As the previous sections made clear, each ontology language exhibits advantages and disadvantages and thus the choice of an ontology language should be driven by the purpose what for it will be used. Of course, as can be seen from the research, OWL claim to be the most popular, having the biggest community and the most spreading and applicable semantic Web language. However, the main drawback of OWL is lack of supporting tools. At this moment we found only two tools – *Protégé* and *OWL Validator* – supporting OWL and allowing to develop OWL ontologies. While, KIF and RDF + RDF(S) having disadvantages, like heavyweight in the first case and lack of affection inference mechanism and the lack of formal underlying model semantics in the second case, they have the broader application. I.e. KIF is used as a basis in SUMO (SUO-KIF language), Protégé PAL. RDF is used as a basis in OWL, DAML, Protégé and FOAF[18].

DBpedia[4] uses the RDF as a flexible data model for representing extracted information and for publishing it on the Web. SPARQL[19] – a query language for RDF – used to query DBpedia data.

Recently, WordNet[7] [36] has been adopted in the Semantic Web research community. It is used mainly for *annotation* and *retrieval* in different domains, like cultural heritage [37] and product catalogs [38], to *ground other vocabularies*, like FOAF, and as *background knowledge* in ontology alignment tools and other applications[20]. Therefore, a number of conversions of WordNet (e.g. WordNet's Prolog format) in RDF and/or OWL [39], [40], [41] are created.

Another similarity used in all languages is a *layered approach* and the use of *standards*. A number of languages have been developed on the basis of a particular standard. For example, RDF is used for developing OWL, XML is used for developing RDF, etc. The main advantage of using a layered approach for creating a particular language is improving and refining existing language. For example, RDF is improved and extended to create OWL.

Recently, a *sublanguage* or *module approach* is also popular. It means that language consists of several modules or sublanguages, where each sublanguage complements the previous sublanguage. This statement can be defined by equation (1).

$$L = \{L_i,\ i = 1, ..., n,\ L_i < L_{i+1}\}, \tag{1}$$

where L is a language consisting of sublanguages L_i and each subsequent sublanguage complements a preceding sublanguage.

This approach is used in OWL, which consists of OWL Lite, OWL DL and OWL Full, and DAML+OIL. Another example is SWRL, allowing users to write rules in terms of OWL concepts. It is created to provide more powerful deductive reasoning capabilities than OWL alone. Semantically, SWRL is built on the same description logic foundation as OWL and provides similar strong formal guarantees when performing inference.

---

[18] http://xmlns.com/foaf/spec/

[19] http://www.w3.org/TR/rdf-sparql-query

[20] http://en.wikipedia.org/wiki/WordNet

The main advantage of the sublanguage approach is that an existing language can be extended by creating a sublanguage without necessity to define a new language with a set of new constructs.

Coding examples from the literature have been presented for all languages. The presented research can be refined and extended according to new ontology languages and criteria of comparison.

References

1. Berners-Lee, T.: The Semantic Web as a language of logic (2009), `http://www.w3.org/DesignIssues/Logic.html`
2. Gutierrez-Pulido, J.R., Ruiz, M.A.G., Herrera, R., Cabello, E., Legrand, S., Elliman, D.: ntology languages for the semantic web: A never completely updated review. Knowl.-Based Syst., 489–497 (2006)
3. Guarino, N.: Formal Ontology and Information Systems. In: Proc. of FOIS 1998, Trento, Italy, pp. 3–15. IOS Press, Amsterdam (1998)
4. OntologyDefinition Metamodel. Object Management Group, OMG (2009), `http://www.omg.org/spec/ODM/`
5. Kalibatiene, D., Vasilecas, O.: Ontology Axioms for the Implementation of Business Rules. Technological and Economic Development of Economy 16(3), 471–486 (2010)
6. Vasilecas, O., Kalibatiene, D., Guizzardi, G.: Towards a Formal Method for Transforming Ontology Axioms to Application Domain Rules. Information Technology and Control 38(4), 271–282 (2009)
7. de Almeida Falbo, R., Silva de Menezes, C., da Rocha, A.R.C.: A Systematic Approach for Building Ontologies. In: Coelho, H. (ed.) IBERAMIA 1998. LNCS (LNAI), vol. 1484, pp. 349–360. Springer, Heidelberg (1998)
8. Culmone, R., Rossi, G., Merelli, E.: An Ontology Similarity algorithm for BioAgent. In: Ercolani, S., Zamboni, M.A. (eds.) NETTAB Workshop on Agents and Bioinformtics, Bologna (2002), `http://www.bioagent.net/WWWPublications/Download/NETTAB02P1.pdf`
9. Lin, S., Miller, L.L., Tsai, H., Xu, J.: Integrating a Heterogeneous Distributed Data Environment with a Database Specific Ontology. In: Sha, E.H.M. (ed.) ISCA PDCS, pp. 430–435 (2001)
10. McGuinness, D.L., Patel-Schneider, P.F.: Usability Issues in Knowledge Representation Systems. In: Mostow, J., Rich, C. (eds.) AAAI/IAAI, pp. 608–614 (1998)
11. Noy, N.F., Fergerson, R.W., Musen, M.A.: The Knowledge Model of Protégé 2000: Combining Interoperability and Flexibility. In: Dieng, R., Corby, O. (eds.) EKAW 2000. LNCS (LNAI), vol. 1937, pp. 17–32. Springer, Heidelberg (2000)
12. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5, 199–220 (1993)
13. Ontology language. Wikipedia (2010), `http://en.wikipedia.org/wiki/Ontology_language`
14. Lavbic, D., Krisper, M.: Facilitating Ontology Development with Continuous Evaluation. Informatica, Lith. Acad. Sci., 533–552 (2010)
15. Labic, D., Krisper, M.: Rapid Ontology Development. In: EJC, pp. 283–290 (2009)
16. Su, X., Ilebrekke, L.: A Comparative Study of Ontology Languages and Tools. In: Pidduck, A.B., Mylopoulos, J., Woo, C.C., Ozsu, M.T. (eds.) CAiSE 2002. LNCS, vol. 2348, pp. 761–765. Springer, Heidelberg (2002)

17. Genesereth, M.R.: Knowledge Interchange Format (KIF). Draft proposed American National Standard (dpANS), NCITS.T2/98-004. Stanford University (2004), http://logic.stanford.edu/kif/dpans.html
18. Patil, R.S., Fikes, R.E., Patel-Schneider, P.F., Mckay, D., Finin, T., Gruber, T., Neches, R.: An Overview of the DARPA Knowledge Sharing Effort. In: Huhns, M.N., Singh, M.P. (eds.) KR 1992, pp. 243–254. Morgan Kaufmann Publishers Inc., San Francisco (1992)
19. Berendt, B., Hotho, A., Mladenič, D., van Someren, M., Spiliopoulou, M., Stumme, G.: A Roadmap for Web Mining: From Web to Semantic Web. In: Berendt, B., Hotho, A., Mladenič, D., van Someren, M., Spiliopoulou, M., Stumme, G. (eds.) EWMF 2003. LNCS (LNAI), vol. 3209, pp. 1–22. Springer, Heidelberg (2004)
20. Antoniou, G., Harmelen, F.V.: Web Ontology Language: OWL. Handbook on Ontologies, 67–92 (2004)
21. Drummond, N., Horridge, M., Stevens, R., Wroe, C., Sampaio, S.: Pizza ontology. The University of Manchester (2007), http://www.co-ode.org/ontologies/pizza/2007/02/12/pizza.owl
22. Schwitter, R.: A Controlled Natural Language Layer for the Semantic Web. In: Zhang, S., Jarvis, R.A. (eds.) AI 2005. LNCS (LNAI), vol. 3809, pp. 425–434. Springer, Heidelberg (2005)
23. Kothari, C.R., Russomanno, D.J.: Relation Elements for the Semantic Web. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) ICCS 2004. LNCS (LNAI), vol. 3127, pp. 275–286. Springer, Heidelberg (2004)
24. Broekstra, J., Klein, M.C.A., Decker, S., Fensel, D., Harmelen, F.V., Horrocks, I.: Enabling knowledge representation on the Web by extending RDF Schema. Computer Networks, 609–634 (2002)
25. Fensel, D., Harmelen, F.V., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F.: OIL: An Ontology Infrastructure for the Semantic Web. IEEE Intelligent Systems, 38–45 (2001)
26. Jong-Soo, S.: Ontology Languages. Korea University (2010), http://onto.tistory.com/attachment/4974544093158B0.ppt
27. Broekstra, J., Klein, M., Decker, S., Fensel, D., Horrocks, I.: Adding formal semantics to the Web building on top of RDF Schema. In: ECDL 2000 Workshop on the Semantic Web (2000), http://www.cs.vu.nl/~mcaklein/presentations/2001-05-02-WWW10-HongKong.pdf
28. Horrocks, I.: DAML+OIL: a Description Logic for the Semantic Web. IEEE Data Eng. Bull., 4–9 (2002)
29. Hendler, J.A.: Agents and the Semantic Web. IEEE Intelligent Systems, 30–37 (2001)
30. Martin, P.: Knowledge representation in CGLF, CGIF, KIF, frame-CG and formalized-english. In: Priss, U., Corbett, D.R., Angelova, G. (eds.) ICCS 2002. LNCS (LNAI), vol. 2393, pp. 77–91. Springer, Heidelberg (2002)
31. Tonti, G., Bradshaw, J.M., Jeffers, R., Montanari, R., Suri, N., Uszok, A.: Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei, and Ponder. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 419–437. Springer, Heidelberg (2003)
32. Lara, R., Roman, D., Polleres, A., Fensel, D.: A Conceptual Comparison of WSMO and OWL-S. In: Zhang, L.-J., Jeckle, M. (eds.) ECOWS 2004. LNCS, vol. 3250, pp. 254–269. Springer, Heidelberg (2004)
33. Krogstie, J., Soelvberg, A.: Information systems engineering – concepttual modeling in a quality perspective. Norwegian University of Technology and Science (1999)

34. McGuinness, D.L., van Harmelen, F. (eds.): OWL Web Ontology Language Overview. W3C Recommendation (2004), http://www.w3.org/TR/owl-features/
35. Horrocks, I.: OWL: A Description Logic Based Ontology Language. In: Gabbrielli, M., Gupta, G. (eds.) ICLP 2005. LNCS, vol. 3668, pp. 1–4. Springer, Heidelberg (2005)
36. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. Bradford Books (1998)
37. Hollink, L., Schreiber, A.T., Wielemaker, J., Wielinga, B.J.: Semantic annotation of image collections. In: Handschuh, S., et al. (eds.) KCAP 2003, pp. 41–48 (2003)
38. Guarino, N., Masolo, C., Vetere, G.: Ontoseek: Content-based access to the web. IEEE Intelligent Systems 14(3), 70–80 (1999)
39. van Assem, M., Gangemi, A., Schreiber, G.: Conversion of WordNet to a standard RDF/OWL representation. In: LREC 2006 (2006)
40. van Assem, M., Gangemi, A., Schreiber, G. (eds.): RDF/OWL Representation of WordNet. W3C Working Draft (2006), http://www.w3.org/TR/wordnet-rdf/
41. WordNet in RDF/OWL, Best Practice (2006), http://ebiquity.umbc.edu/blogger/2006/02/20/wordnet-in-rdfowl-best-practice/