Available online at www.sciencedirect.com
ScienceDirect
Knowledge-Based Systems 19 (2006) 489–497
Knowledge-Based SYSTEMS
www.elsevier.com/locate/knosys

Ontology languages for the semantic web: A never completely updated review

J.R.G. Pulido [a,*], M.A.G. Ruiz [b], R. Herrera [c], E. Cabello [d], S. Legrand [e], D. Elliman [f]

[a] *Faculty of Telematics, University of Colima, México*
[b] *Virtual Reality Laboratory, University of Colima, México*
[c] *SIABUC Dept, University of Colima, México*
[d] *Info Systems and Computing Department, University of Valencia, Spain*
[e] *Department of Computer Science, University of Jyväskylä, Finland*
[f] *Computer Science and IT School, University of Nottingham, UK*

Received 10 March 2006; accepted 30 April 2006
Available online 23 June 2006

Abstract

This paper gives a never completely account of approaches that have been used for the research community for representing knowledge. After underlining the importance of a layered approach and the use of standards, it starts with early efforts used for artificial intelligence researchers. Then recent approaches, aimed mainly at the semantic web, are described. Coding examples from the literature are presented in both sections. Finally, the semantic web ontology creation process, as we envision it, is introduced.

Keywords: Semantic web; Ontology languages; Knowledge representation; Ontology creation process

1. Introduction

In recent years, several markup languages have been developed for realizing the semantic web. The construction of these languages is evolving according to a layered approach to language development, in particular at the level of the *ontology vocabulary* (Fig. 1 from [1]) as it is in this layer where the basis to carry out reasoning and inferencing are laid. These languages must meet a number of requirements. They[1] must [2]:

- Have a compact syntax.
- Be highly intuitive to humans.
- Have a well-defined formal semantics.
- Be able to represent human knowledge.
- Include reasoning properties.
- Have the potential for building knowledge bases.
- Have a proper link with existing web standards to ensure interoperability.

Unlike some existing markup languages, specifically HTML, a semantic web 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. Because the web is decentralized, the language must also allow for the definition of diverse vocabularies and let them evolve. Some existing languages let authors create ontologies by defining class taxonomies and relationships between multiple classes. Some other also allow the formation of more complex definitions by using axioms from some form of logic. The idea in this context is to add ontology-based metadata to web pages and improve accessibility providing a means for reasoning about content [45–48].

* Corresponding author. Tel./fax: +52 312 316 1075.
E-mail address: jrgp@ucol.mx (J.R.G. Pulido).
[1] See also http://www.w3.org/DesignIssues/Logic.html.

0950-7051/$ - see front matter
doi:10.1016/j.knosys.2006.04.013

2. Early approaches

In this section some early languages for representing knowledge are briefly discussed, namely, the Knowledge Interchange Format, F-Logic, the Dublin Core, and The CYC project.

2.1. Knowledge Interchange Format

The Knowledge Interchange Format (KIF) is a formal language for the interchange of knowledge among disparate computer programs. The following are some of its features [3]:

- Declarative semantics. It is possible to understand the meaning of expressions in the language without appealing to an interpreter for manipulating the expressions.
- Logically comprehensive. It provides for the expression of arbitrary sentences in predicate calculus.
- Metaknowledge. This allows us to make all knowledge representation explicit and permit us to introduce new knowledge representation constructs without changing the language.
- Translatability. It enables practical means of translating declarative knowledge bases to and from typical knowledge representation languages.
- Readability. Although KIF is not intended as a language for interaction with humans, it is useful for describing representation language semantics and assisting humans with knowledge base translation problems.

As any declarative representation language, it requires a conceptualization of the world in terms of objects, functions, and relations. KIF is a language that was developed by the interlingua working group under the DARPA knowledge sharing initiative to facilitate knowledge sharing. It was designed to be a *state-of-the-art* interlingua tool. KIF is an extended version of first-order predicate calculus, and essentially an intermediary language for translating different knowledge representation languages. Its specifications are meant to be sharable. The sentence *All writers are misunderstood by some reader* is shown in Table 1 as a KIF sentence [4].

2.2. F-Logic

F-Logic is a full-fledged logic that includes a model-theoretic semantics and a sound and complete proof theory. This makes it computationally attractive and renders it as a suitable basis for developing a theory for object-oriented logic programming. F-Logic is an integration of frame-based languages and first-order predicate calculus. It includes objects, inheritance, polymorphic types, query methods, and encapsulation. Its deductive system works with the theory of predicate calculus and structural and behaviour inheritance [5]. It is capable of representing virtually all aspects of the object-oriented paradigm. Its main achievement is to integrate conceptual modelling constructs into a coherent logical framework. It provides classes, attributes with domain and range definitions, *is–a* hierarchies with set inclusion of subclasses, and logical axioms between elements of an ontology and its instances. Table 2 shows an example of F-Logic declarations [5].

Table 1
KIF example

```
(forall ?w
   (⇒(writer ?w)
      (exists (?r ?d)
        (and (reader ?r) (document d?)
            (writer ?w ?d) (read ?r ?d)
          (not (understands ?r ?d))
        )
      )
   )
)
```

Table 2
F-Logic example

```
bob: manager
1989: year
manager::empl
mary: faculty
10000: int
.
.
.
faculty :: empl
mary[boss→bob]
empl [boss⇒manager; salary @ year⇒integer]
faculty[boss⇒faculty]
```

2.3. Dublin core

The oldest and most widely adopted initiative for global markup is the Dublin[2] Core (DC). Its goal is to facilitate electronic resource discovery on the web. It consists of a set of 15 elements for describing web resources, and it is the *de facto* worldwide standard for information resources across disciplines and languages [6]. It has already been translated into 25 languages. The DC is a metadata element set for describing cataloguing information, such as that needed in digital libraries. This initiative early embraced RDF as the framework on which to build such metadata [7]. Simplicity is both the strength and the weakness of it. The initial aim was to create a single set of metadata elements for untrained people who publish electronic materials for describing their work. Some people continue to hold this minimalist view, a simple set of rules that anyone can apply. Others prefer the benefits that come from more tightly controlled cataloguing rules and would accept the additional labour and cost. Table 3 shows a DC example coded in HTML [8].

2.4. CYC

This[3] knowledge base is a formalized representation of a vast quantity of fundamental human knowledge: facts,

[2] http://dublincore.org.
[3] http://www.cyc.com.

Table 3
Dublin Core example

```
<meta name=''DC.subject''
content=''Dublin Core Metadata Element Set''>
<meta name=''DC.publisher'' content='' OCLC
  online
computer library center, inc.''>
.
.
.
<meta name=''DC.title'' content='' Dublin
  Core Element
Set Reference Page''>
<meta name=''DC.identifier'' scheme='' URL''
content=''http://purl.oclc.org/metadata/
  dublin_core">
```

rules of thumb, and heuristics for reasoning about objects and events of everyday life. The initial aim of this project was to specify a large common-sense ontology that should provide artificial intelligence to machines. Far from having attained its goal, CYC still provides the worldwide largest formalized ontology. It provides formal axiomating theories for many aspects of common-sense knowledge for developing ontologies for a wide variety of specific domain applications.

CycL is a declarative and expressive language similar to first-order predicate calculus with extensions. It uses a form of circumscription, includes unique names, and can make use of a classed world assumption when appropriate. It has an inference engine to perform several kinds of reasonings. Hundreds of thousands of concepts have been formalized with millions of logical axioms, which are known as microtheories, that specify constraints of the individual objects and classes. Each microtheory captures the knowledge and reasoning required for some particular domain. CYC is not a monolithic integrated ontology, but a network of microtheories for a set of domains whose union covers the different ontological commitments [50] that can be made within those domains. Table 11 shows a CYC sentence taken from its natural[4] language processing system.

3. Recent approaches

In this section some recent ontology languages, particularly useful for the semantic web, are presented, namely, the extended markup language, the resource description framework, the knowledge annotation initiative, the simple HTML ontology extensions, the ontology interchange language, and the DARPA Agent Markup Language.

3.1. The eXtended Markup Language

The eXtended Markup Language (XML) was the first language to separate the markup of web content from web presentation, facilitating the representation of task-specific and domain-specific data on the web. Unfortunately it lacks semantics. Computer agents cannot be guaranteed to determine the intended interpretation of its tags. It is designed to describe the structure of a document, not the content. XML includes a Document Type Definition (DTD) which is used to enforce constraints on which tags to use and how they should be nested within a document. A DTD defines a grammar to specify allowable combinations and nestings of tag names and attribute names. The DTD specifies only semantic conventions, not any semantics. For DTDs are the closest component that XML offer for ontological modelling, it is easy to consider them as a simple ontology mechanism, however some differences between DTDs and ontologies exist [9,10]:

- A DTD specifies the legal lexical nesting of a document, which may or may not coincide with an ontological hierarchy. An *is–a* relationship does not exist in XML for instance.
- Attributes are no longer local to a concept, they are global to a document, at least for the representation of ontology attributes as XML elements.
- DTDs have no notion of inheritance. In an ontology, subclasses inherit at tributes from their superclasses.
- DTDs define the order in which tags appear in a document. In an ontology the order does not matter.
- Ontologies have a much richer means of defining semantics. The lack of expressivity of DTD prohibits the formulation of axioms.

XML is widely known in the internet community and has been used as a basis for a number of software development activities [11–13]. A well-formed XML document creates a balanced tree of nested sets of open and close tags, each of which may include several *attribute–value* pairs. Table 4 shows an XML example.

3.2. The Resource Description Framework

The Resource Description Framework (RDF) is a standard for the web metadata that the World Wide Web Consortium (W3C) developed. It is suitable for describing any web resource and as such it provides interoperability between applications that exchange machine-understandable information on the web. RDF is becoming a widely recognized language and a representation formalism that can serve as a worldwide interlingua for information interchange. The RDF description model uses *object–attribute–*

Table 4
XML example

```
<?XML version=''1.0''?>
<book>
<author>M Goossens</author>
<title>The LaTeX companion</title>
<publisher>Addison-Wesley, Reading, Mass</
  publisher>
<year>1994</year>
</book>
```

[4] http://www.cyc.com/cycdoc/ref/nl.html.

value triples, also known as *statements* (Table 5). 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. It provides a means of describing the relationships among resources in terms of the named properties and values. RDF has significant advantages over XML. The object-attribute structure provides natural semantic units because all objects are independent entities. The RDF model can still be used even if XML's syntax changes or disappears because RDF describes an independent layer.

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 when using XML end up building a layer on the top of XML to integrate these ontological primitives. RDFS lets developers define a particular vocabulary for RDF data and specify the kind of object to which these attributes may be applied. This mechanism provides a basic type system for RDF models and interpretation of RDF expressions. RDF played an important role as a basis for DARPA Agent Markup Language (DAML), discussed later in this section, whose layers of logic are to be built on the top of the basic RDF framework [14,44].

3.3. The knowledge annotation initiative of knowledge acquisition

This initiative, also known as $(KA)^2$, was a case study on the process of populating a shared ontology for a heterogeneous and world-wide research community. The use of an ontology for providing semantic access to on-line information sources of this community was tested. It comprises three main tasks [15,49]:

- Ontological engineering to build an ontology of the subject matter. The design criteria used were: modularity, specialization, classification of concepts, and standardized names.
- Characterising the knowledge in terms of the ontology. Each participant had to annotate relevant knowledge on their web pages by using the *onto* attribute (Table 6). An annotating tool was provided.
- Delivering intelligent access to the knowledge. A tool for querying the knowledge using F-logic formulations and a graphical user interface for non-expert users were also provided.

Table 5
RDF example

```
bookName{http://www.amazon.co.uk/exec/ obidos/
  ASIN
/0198538642/202-6666526-7532666,
''Neural Networks for Pattern Recognition''}

publisher{http://www.amazon.co.uk/exec/obidos/ASIN
/0198538642/202-6666526-7532666, ''Clar endon
  Press''}

numberOfPages{http://www.amazon.co.uk/ exec/obi-
dos/ASIN
/0198538642/202-6666526-7532666, ''500''}
```

Table 6
$(KA)^2$ example

```
<html><head><a ONTO=''page :Researcher''/>
<title>Richard Benjamins</title>
</head><body><h1>
<a ONTO=''page[firstName=body]''>Richard</a>
<a ONTO=''page[lastName=body]''>Benjamins</a>
<a ONTO=''page[affiliation=body]''>
Artificial Intelligence Research Institute
(IIIA)</a>
<a href="http://www.iiia.csic.es">CSIC</a>,
Barcelona, Spain.
</h1>
.
.
.
</body></html>
```

Several researchers cooperated to construct the ontology. *Ontobroker* is the intelligent agent that provides access to the knowledge.

3.4. Simple HTML Ontology Extensions

Giving the authors the ability to embed knowledge directly into HTML pages, making it also simple for user-agents and robots to retrieve and store knowledge, was the goal of the so-called *Simple HTML Ontology Extension* (SHOE). This approach allows authors to add semantic content to web pages, relating the context to common ontologies that provide contextual information about the domain [16]. Most web pages with *SHOE* annotations tend to have tags that categorize concepts, therefore there is no need for complex inference rules to perform automatic classification [17]. This approach extends HTML with a set of object-oriented tags to provide structure for knowledge acquisition. It associates meaning with content by committing web pages to existing ontologies. These ontologies permit the discovery of implicit knowledge through the use of taxonomies and inference rules, allowing information providers to encode only the necessary information into their web pages. An *ontology* tag delimits the machine-readable portion of the ontology. Some other tags[5] complement the definition of ontologies (Table 7). SHOE focuses on the problem of maintaining consistency as the ontologies evolve. In [18] the use of SHOE in a real world internet application is described. Tools for annotating pages, information gathering tasks, and querying are provided.

[5] See http://www.cs.umd.edu/projects/plus/SHOE/ontologies.html.

Table 7
SHOE example

```
<html><body><ontology>
<USE-ONTOLOGY ID=''base-ontology''
  VERSION=''1.0''
PREFIX=''base''
URL="http://www.cs.umd.edu/projects/ plus/SHOE/
  base.html">
.
.
.
<DEF-CATEGORY NAME=''Department'' ISA=''
  Organization''>
<DEF-RELATION NAME=''advisor''>
   <DEF-ARG POS=''1'' TYPE=''Student''>
   <DEF-ARG POS=''2'' TYPE=''Professor''>
</DEF-RELATION>
</ontology></body></html>
```

3.5. The Ontology Interchange Language

The Ontology Interchange Language (OIL) is based on three elements, namely, frame-based systems, description logics, and web standards [2,19].

- The central modelling primitives of frame-based systems are frames with properties. These properties have a local scope and are only known to the frames for which they have been defined. A frame provides a certain context for modelling one aspect of a domain. OIL is based on the notion of a class and its superclasses and attributes. Relations can be defined as independent entities.
- Description[6] logics have been developed in knowledge representation research for describing knowledge in terms of concepts and roles. 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. Table 8 shows an example defining African wildlife [20].
- Given the importance of the WWW, ontology languages must be developed bearing web standards in mind. OIL has a well-defined syntax in XML. It is also defined as an extension of the RDF and its extension schema (RDFS), which provides two important contributions: a standardized syntax for writing ontologies and a standard set of modelling primitives.

3.6. The DARPA Agent Markup Language

The DARPA Agent Markup Language (DAML) is a US Government-sponsored endeavour aimed at providing the foundation for the next web evolution, the semantic web. Academic researchers, government agencies, software development companies, and industrial organizations are participating in the program [7]. 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. Table 9 shows an example taken from [21].

Table 8
OIL example

```
class-def animal
class-def plant subclass-of NOT animal
class-def tree subclass-of plant
.
.
.
class-def defined herbivore subclass-of animal
   NOT carnivore
   slot-constraint eats value-type plant
   OR (slot-constraint is-part-of has- value
     plant)
class-def giraffe subclass-of herbivore
   slot-constraint eats value-type leaf
```

Table 9
DARPA example

```
<daml:Class>
<daml:intersectionOf
  rdf:parseType=''daml:collection''>
<daml:Class rdf:about=''#Human''/>
<daml:Class rdf:about=''#Male''/>
</daml:intersectionOf>
</daml:Class>
.
.
.
<daml:Restriction daml:minCardinalityQ=''2''>
<daml:onProperty rdf:resource=''#hasChild''/>
<daml:hasClassQ rdf:resource=''#Lawyer''/>
</daml:Restriction>
```

Building on the top of RDF and RDFS, and with its root in description logics, 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, thus making it amenable to agent interoperability and automated-reasoning techniques. The Inference Language (DAML-L) is a logical language with a well-defined semantics and the ability to express at least propositional Horn clauses, which enable compact representation of constraints and rules for reasoning. The language ties the information on a page to machine-readable semantics and allows for communities to extend simple ontologies for their own use. In addition, it provides mechanisms for the explicit representation of services, processes, and business models, so as to allow non-explicit information to be recognized [1]. DAML + OIL and DAML-L together provide a markup language for the semantic web with expressive power and a well-defined semantics for reasoning. The DAML family of markup languages enable web service providers to develop semantically grounded, rich representations of web services that a number of different agent

[6] They result from early work on *semantic networks* and first-order logic. They allow to define efficient inference procedures. Papers, projects, and research in this area can be found at http://dl.kr.org.

Table 10
OWL example

```
<owl:Class>
  <owl:oneOf rdf:parseType=''Collection''>
    <owl:Thing rdf:about=''#Eurasia''/>
    <owl:Thing rdf:about=''#Africa''/>
    <owl:Thing rdf:about=''#NorthAmerica''/>
    <owl:Thing rdf:about=''#SouthAmerica''/>
    <owl:Thing rdf:about=''#Australia''/>
    <owl:Thing rdf:about=''#Antarctica''/>
  </owl:oneOf>
</owl:Class>
```

Table 11
CYC example

```
#$Department departments
 The collection #$Department is a subset of
 #$Organization. An element of #$Department is a
 major sub-organization of a business, government,
 or academic organization. An element of
 #$Department is part of the organization to which
 it belongs, NOT a separate legal entity (such as a
 partly or wholly owned subsidiary company), and it
 performs some of the activity of that organization.

  direct specialization of: #$Organization
  direct instance of: #$ExistingObjectType
```

architectures and technologies can exploit to a variety of different ends [22].

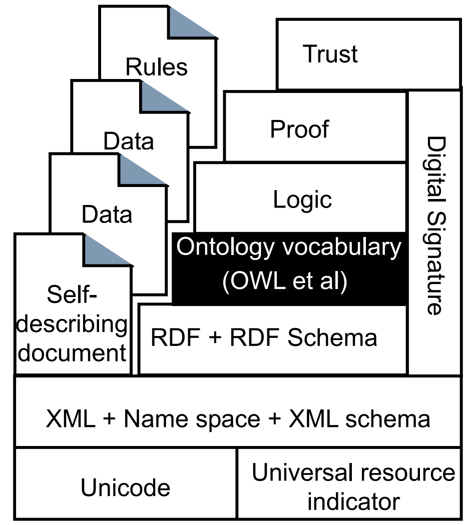
3.7. Ontology Web Language (OWL)

This is the result of some hard-work carried out by some experts and semantic web enthusiasts. It now an standard ontology language for the semantic web. It is compatible with early ontology languages, including SHOE, DAML + OIL, and provides the engineer more power to express semantics. It includes *conjuction, disjunction, exitencially, and universally quantified variables*. Reasoners can make use of this to carry out logical inferences and derive knowledge. Its expressiveness, however, has some drawbacks [23]:

(1) Some constructs are very complex, that is why it comes in three flavors.
(2) Reasoning is not efficient as there is a trade-off against time-complex cost.
(3) It is not easy to use, here is where authoring software tools fit.
(4) It is not intuitive, need to be owl-savvy to build efficient knowledge constructions.

It is not possible to satisfy all of the constraints of a domain, that is why it comes in three flavors [24]:

(1) OWL FULL: for an upward compatibility with RDF both at syntax and semantic level, a legal RDFS schema is also an OWL documents. We also can change RDF primitives.



The focus is on the *ontology vocabulary* Layer (in black).

Fig. 1. Web language layers.

(2) OWL DL: in order to be less time-complex, this version of OWL has been created. It allows efficient reasoning and inferencing but loses backward compatibility with owl full.
(3) OWL Lite : an even more restricted subset of owl full, for an expressive ontology language with decidable inference. Implementers love this version.

Table 10 shows an OWL[7] example. The different versions of this ontology language give implementers choices for them to select the best one depending on their system requirements. This semantic web language is to allow mechanisms to convert the current web into a semantic one. This transition is yet to take a while as software tools that help us in the semi-automatic construction of ontology components for the semantic web become available.

In the following section the elements that are required for the next generation web to become a reality are introduced in turn.

4. Discussion

For the semantic web to become a reality, a number of frameworks have to be built to support the ontology creation activities (Fig. 2) involved in the process. These activities, as we envision this process, are as follows:

Gathering Before the extraction phase, we have to collect documents carrying knowledge from the domain we are interested in, process them, and end with a suitable form to carry out the next operations. It usually involves dealing with unstructured data in natural language from digital archives [25–28,43,51,52]. Some

[7] Taken from *http://www.w3.org/TR/2003/WD-owl-ref-20030331*.

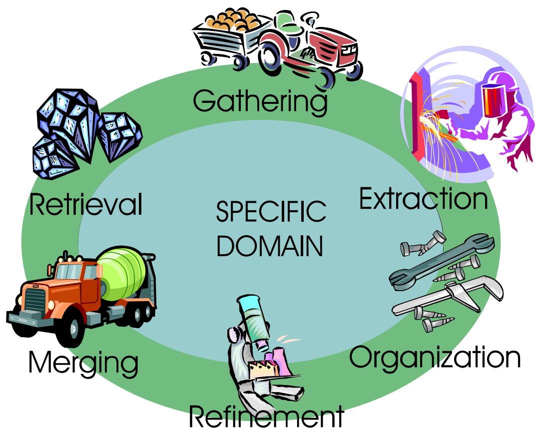


Fig. 2. The semantic web ontology creation process.

useful software tools to carry out gathering tasks are: Spade[8] and OntoExtract[9].

*Extraction*A number of ontology learning tools are available. The purpose of these kind of software is to help the ontology engineer to explore specific domains and extract ontology components. This requires background knowledge for creating taxonomies of the domain in a semi-automatic way. Learning techniques may be applied by the knowledge engineer for this task [29–34]. Some useful extraction software tools are: Grubber[10] and OntoBuilder[11].

Organization Once the ontology components have been extracted from the domain, it is time to generate formal representations of the knowledge acquired. Ontology software tools may be useful at this stage. Later, this knowledge may be embedded into digital archives, e.g., web pages, to be used by software agents or humans [35,36,2,37]. Some useful ontology software tools are: OntoEdit[12], SMORE[13], and Protégé[14].

Merging Defining mapping rules to facilitate interlingua exchange relating information from one context to another. This activity is as important as *Extraction*. It can be referred to as finding commonalities between two knowledge bases and deriving a new knowledge base [38,31,39]. Some helpful software tools for merging ontologies are: PROMPT[15], and quimaera[16].

Refinement Improving the structure and content of the knowledge by eliciting knowledge from the domain experts. It amends the knowledge at a finer granularity level. It is also of particular importance after *merging* operations (cf. [38,40–42]), for instance, when two e-commerce agents are trying to negotiate. A number of software tools for organizing ontology components include refinement capabilities as well.

Retrieval This is the ultimate semantic web goal and it is going to take a while yet before we see smart software applications, but when the semantic web is populated, then those applications, e.g., semantic robots, agents, will traverse the web looking for data for us in a knowledge-based fashion. In the mean while, we still have to wait for those frameworks to mature. Racer[17], and KAON2[18] are some promising early tools to carry out these tasks.

This paper has given a never completely account of approaches that have been used for the research community for representing knowledge. After underlining the importance of a layered approach and the use of standards, it started with early efforts used for artificial intelligence researchers. Then recent approaches, aimed mainly at the semantic web, were described. Coding examples from the literature have been presented for both sections. Finally, the semantic web ontology creation process, as we envision it, has been introduced.

8 http://venus.ucol.mx/~jrgp/en/ol-approach.html.
9 http://www.ontoknowledge.org/tools/toolrep.shtml#Individual_tools.
10 http://venus.ucol.mx/~jrgp/en/ol-approach.html.
11 http://iew3.technion.ac.il/OntoBuilder.
12 http://www.ontoknowledge.org/tools/ontoedit.shtml.
13 http://www.mindswap.org/2005/SMORE.
14 http://protege.stanford.edu.
15 http://protege.stanford.edu/plugins/prompt/prompt.html.
16 http://www.ksl.Stanford.EDU/software/chimaera.
17 http://www.sts.tu-harburg.de/~r.f.moeller/racer.
18 http://kaon2.semanticweb.org.

References

[1] J. Hendler, Agents and the semantic web, IEEE Intell. Syst. 16 (2) (2001) 30–37.

[2] D. Fensel, et al., OIL: an ontology infrastructure for the semantic web, IEEE Intell. Syst. 16 (2001) 38–45.

[3] M. Genesereth, R. Fikes, Knowledge interchange format, version 3.0 reference manual, Tech. Rep. KSL-92-86, Knowledge Systems Laboratory, Stanford University, 1992.

[4] T. Gruber, A translation approach to portable ontology specifications, Tech. Rep., Knowledge Systems Laboratory, Stanford University, 1993.

[5] M. Kifer, et al., Logical foundations of object-oriented and frame-based languages, J. ACM 42 (4) (1995) 741–843.

[6] M. Tallis, et al., The briefing associate: easing authors into the semantic web, IEEE Intell. Syst. 17 (1) (2002) 26–32.

[7] O. Lassila, et al., The semantic web and its languages, IEEE Intell. Syst. 15 (6) (2000) 67–73.

[8] W. Arms, Digital Libraries and Electronic Publishing Series, MIT Press, USA, 2000.

[9] M. Erdmann, R. Studer, How to structure and access XML documents with ontologies, Data Knowl. Eng. 36 (2001) 317–335.

[10] D. Fensel, Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Springer-Verlag, Berlin, 2000.

[11] R. Mello, et al., A method for the unification of XML schemata, Inf. Softw. Technol. 44 (2002) 241–249.

[12] R. Luke, et al., A survey in indexing and searching XML documents, J. Am. Soc. Inf. Sci.Technol. 53 (6) (2002) 415–437.

[13] P. Haase, et al., A comparison of RDF query languages, in: The Semantic Web ISWC 2004: Third International Semantic Web Conference, in: S. McIlraith (Ed.), Lecture Notes in Computer Science (LNCS), Springer, 2004, pp. 502–517.

[14] S. Staab, et al., AI for the web—ontology-based community web portals, Comput. Netw. 33 (2000) 473–491.
[15] R. Benjamins, et al., $(KA)^2$: building ontologies for the internet: a midterm report, Int. J. Hum.-Comput. Stud. 51 (3) (1999) 687–712.
[16] S. Luke, et al., Ontology-based web agents, in: First International Conference on Autonomous Agents, ACM, New York, 1997, pp. 59–66.
[17] J. Heflin, J. Hendler, Dynamic ontologies on the web, in: American Association For Artificial Intelligence Conference, AAAI Press, California, 2000, pp. 251–254.
[18] J. Heflin, et al., Applying ontology to the web: a case study, Engineering Applications of Bio-Inspired Artificial Neural Networks 1607.
[19] D. Fensel, et al., OIL in a nutshell, in: R. Ding (Ed.), Proceedings of the European Knowledge Acquisition Conference, Lecture Notes in Artificial Intelligence (LNAI), Springer-Verlag, Berlin, 2000.
[20] S. Decker, et al., The semantic web: the roles of XML and RDF, IEEE Internet Comput. 4 (5) (2000) 63–74.
[21] I. Horrocks, DAML + OIL: a reasonable web ontology language, in: Lecture Notes in Computer Science (LNCS), Vol. 2287, Springer-Verlag, Berlin, 2002, pp. 2–13.
[22] S. McIlraith, et al., Semantic web services, IEEE Intell. Syst. 16 (2) (2001) 46–53.
[23] B. Berent, et al., A roadmap for web mining: from web to semantic web, in: First European Web Mining Forum (EWMF), in: B. Berent (Ed.), Lecture Notes in Computer Science (LNCS), Springer, 2004, pp. 1–22.
[24] I. Horrocks, et al., From SHIQ and RDF to OWL: the making of a web ontology language, J. Web Semantics. 1 (1) (2003) 7–26.
[25] S. Galicia, et al., Web-based sources for a annotated corpus building and composite proper name identification, in: Advances in Web Intelligence: Second International Atlantic Web Intelligence Conference, AWIC 2004, in: J. Favela (Ed.), Lecture Notes in Computer Science (LNCS), Vol. 3034, Springer, 2004, pp. 115–124.
[26] C. Bădica, A. Bădica, Rule learning for feature values extraction from HTML product information sheets, in: G. Antoniou, H. Boley (Eds.), Rules and Rule Markup Languages for the Semantic Web: Third International Workshop, RuleML 2004, Lecture Notes in Computer Science (LNCS), Vol. 3323, Springer, 2004, pp. 37–48.
[27] D. Mladenić, M. Grobelnick, Mapping documents onto web page ontology, in: B. Berendt (Ed.), Web Mining: From Web to Semantic Web: First European Web Mining Forum, EWMF 2003, Lecture Notes in Computer Science (LNCS), Vol. 3209, 2004, pp. 77–96.
[28] D. Elliman, *JRG Pulido*, Visualizing ontology components through self-organizing maps, in: D. Williams (Ed.), Sixth International Conference on Information Visualization (IV02), London, UK, IEEE Computer Soc. Press, Los Alamitos, 2002, pp. 434–438.
[29] T. Quan, et al., Automatic generation of ontology of scholarly semantic web, in: S. McIlraith (Ed.), The Semantic Web ISWC 2004: Third International Semantic Web Conference, Lecture Notes in Computer Science (LNCS), Vol. 3298, Springer, 2004, pp. 726–740.
[30] S. Legrand, *JRG Pulido*, A hybrid approach to word sense disambiguation: Neural clustering with class labeling, in: P. Buitelaar, et al. (Eds.), Workshop on Knowledge Discovery and Ontologies, 15th European Conference on Machine Learning (ECML) and 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), Pisa, Italy, 2004, pp. 127–132.
[31] M. Ehrig, S. Staab, QOM—quick ontology mapping, in: S. McIlraith (Ed.), The Semantic Web—ISWC 2004: Third International Semantic Web Conference, Lecture Notes in Computer Science (LNCS), Vol. 3298, Springer, 2004, pp. 683–697.
[32] A. Maedche, V. Zacharias, Clustering ontology-based metadata in the semantic web, in: T. Elomaa (Ed.), Principles of Data Mining and Knowledge Discovery: Sixth European Conference, PKDD 2002, Lecture Notes in Computer Science (LNCS), Vol. 2431, Springer, 2002, pp. 348–360.
[33] A. Maedche, S. Staab, Ontology learning for the semantic web, IEEE Intell. Syst. 16 (2) (2001) 72–79.
[34] D. Elliman, *JRG Pulido*, Self-organizing maps for detecting ontology components, in: H. Arabnia, et al. (Eds.), The 2003 International Conference on Artificial Intelligence (IC-AI), Las Vegas, USA, CSREA Press, 2003, pp. 650–653.
[35] M. Vargas, et al., MnM: Ontolgy driven semi-automatic and automatic support for semantic markup, in: A. Gómez, V. Benjamins (Eds.), Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web: 13th International Conference, EKAW 2002, Lecture Notes in Computer Science (LNCS), Vol. 2473, Springer, 2002, pp. 379–391.
[36] J. Goldbeck, et al., New tools for the semantic web, in: A. Gómez, V. Benjamins (Eds.), Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web: 13th International Conference, EKAW 2002, Lecture Notes in Computer Science (LNCS), Vol. 2473, Springer, 2002, pp. 392–400.
[37] J. Fernandez, R. Martinez, A cooperative tool for facilitating knowledge management, Expert Syst. Appl. 18 (2000) 315–330.
[38] J. Euzenat, An API for ontology alignment, in: S. McIlraith (Ed.), The Semantic Web ISWC 2004: Third International Semantic Web Conference, Lecture Notes in Computer Science (LNCS), Vol. 3298, Springer, 2004, pp. 698–712.
[39] R. Eijk, et al., On dynamically generated ontology translators in agent communication, Int. J. Intell. Syst. 16 (2001) 587–607.
[40] N. Sugiura, et al., Towards on-the-fly ontology-construction-focusing on ontology quality improvement, in: C. Bussler, et al. (Eds.), The Semantic Web: Research and Applications: First European Semantic Web Symposium, ESWS 2004, Lecture Notes in Computer Science (LNCS), Vol. 3053, Springer, 2004, pp. 1–15.
[41] A. Maedche, S. Staab, Measuring similarity between ontologies, in: A. Gomez, R. Benjamins (Eds.), Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web: 13th International Conference, EKAW 2002, Lecture Notes in Computer Science (LNCS), Vol. 247, Springer, 2002, pp. 251–263.
[42] M. Hatala, G. Richards, Value-added metatagging: Ontology and rule based methods for smarter metadata, in: M. Schroeder, G. Wagner (Eds.), Rules and Rule Markup Languages for the Semantic Web: Second International Workshop, RuleML 2003, Lecture Notes in Computer Science (LNCS), Vol. 2876, Springer, 2003, pp. 65–80.
[43] F. Ciravegna, et al., Learning to harvest information for the semantic web, in: C. Bussler, et al. (Eds.), The Semantic Web: Research and Applications: First European Semantic Web Symposium, ESWS 2004, Lecture Notes in Computer Science (LNCS), Vol. 3053, Springer, 2004, pp. 312–326.
[44] J. Alferes, et al., Semantic web logic programming tools, in: F. Bry, et al. (Eds.), Principles and Practice of Semantic Web Reasoning: International Workshop, PPSWR 2003, Lecture Notes in Computer Science (LNCS), Springer, Vol. 2901, 2003, pp. 16–32.
[45] S. Staab, et al., Semantic community web portals, Comput. Netw. 33 (2000) 473–491.
[46] S. Staab, A. Maedche, Knowledge portals: ontologies at work, AI Magazine 22 (2) (2001) 63–75.
[47] G. Miller, et al., Introduction to WordNet: An on-line lexical database, Int. J. Lexicogr. 3 (4) (1991) 235–244.
[48] N. Guarino, et al., OntoSeek: content-based access to the web, IEEE Intell. Syst. (1999) 70–80.
[49] R. Benjamins, et al., Knowledge management through ontologies, in: Second International Conference on Practical Aspects of Knowledge Management, 1998.

[50] A. Waterson, A. Preece, Verifying ontological commitment in knowledge-based systems, Knowl.-Based Syst. 12 (1999) 45–54.

[51] D. Elliman, *JRG Pulido*, Automatic derivation of on-line document ontologies, presented at the Int. Workshop on Mechanisms for Enterprise Integration: From Objects to Ontologies (MERIT 2001), the 15th European Conference on Object Oriented Programming, Budapest, Hungary June, 2001.

[52] JRG. Pulido, et al., Identifying ontology components from digital archives for the semantic web, in: IASTED Advances in Computer Science and Technology (ACST), 2006.