An Overview of the Semantic Web
Improving Web Data Accessibility and Performance

Antoine Abou Rjeily[1-2], Pélagie Houngue[1], Joe Tekli[2-3], Richard Chbeir[3], and Kokou Yetongnon[1]
[1] *LE2I Laboratory UMR-CNRS, University of Bourgogne (UB), 21000 Dijon, France*
[2] *Faculty of Engineering, Antonine University (UPA), 40016, Baabda, Lebanon*
[3] *LIUPPA Laboratory, University of Pau (UPPA), 64200, Anglet, France*
antoine_abou-rjeily@etu.u-bourgogne.fr, {pelagie.hongue, kokou.yetongnon}@u-bourgogne.fr,
joe.tekli@upa.edu.lb, richard.chbeir@univ-pau.fr

Keywords: Web, Semantic Web, Knowledge base, data semantics, XML, RDF, OWL, SPARQL, information retrieval.

Abstract: The Internet has known a very fast evolution, going from the Web 1.0, i.e., the traditional Web where users are merely consumers of static information, to the more dynamic Web 2.0, known as the Social or Collaborative Web, where users produce and consume information simultaneously, and heading toward the more sophisticated and eagerly anticipated Web 3.0, better known as the Semantic Web: extending the Web by giving information well defined meaning so that it becomes more easily accessible by human users and automated processes. This paper briefly describes the evolution of the Web towards the Semantic Web (3.0), providing an overview of the various technological breakthroughs contributing to this evolution, covering: knowledge bases and semantic data description, as well as XML-based data representation and manipulation technologies (i.e., RDF, RDFS, OWL, and SPARQL). We also present the main application domains characterizing the Semantic Web, ranging over information retrieval, information extraction, machine translation, content analysis, and lexicography, and discuss some emergent and future directions aiming at improving Web data accessibility and performance.

1 INTRODUCTION

The *Semantic Web* (SW) (Bertails A. *et al.* 2010; Champavère J. 2010; Joo J. 2011) is a collaborative movement guided by the World Wide Web Consortium (W3C), aiming to extend the Web (as we know it) by *giving information well defined meaning*, in order to improve data accessibility for humans and machines (Berners-Lee T. *et al.* 2001; Bertails A. *et al.* 2010). Also known as *Data Web* or *Web 3.0*, the SW is a vision of the Web where machines are able to automatically exploit the semantic meaning of information, available at different locations in a distributed environment, so as to allow more effective and intelligent Web data access, search and retrieval. However, the capture and processing of semantic information is a difficult task because of the well-known problems that machines have with processing semantics. For instance, a machine traditionally processes the expression "*university"* as a word consisting of 10 characters, rather than capturing the meaning of the word: "*an academic institution of higher education, etc."*, unless some sort of semantic data processing is involved. Hence, the SW vision aims to associate machine-readable semantic descriptions to Web data, using two major technological breakthroughs: i) knowledge bases (such as taxonomies and/or ontologies (Resnik P. 1995; Baziz M. *et al.* 2005)), which provide predefined semantic information references (similarly to dictionaries for human users) to allow the identification and extraction of semantic meaning from raw data, and ii) XML-based data representation technologies (namely RDF (Decker S. *et al.* 2000; Hayes P. 2004) and RDFS (Brickley D. and Guha R. V. 2004; Klyne G. *et al.* 2004) for resource description, OWL (Antoniou G. *et al.* 2004; Dean M. *et al.* 2004) for ontology definition, and SPARQL (Prudhommeaux E. *et al.* 2008; Hartig O. *et al.* 2009) for semantic data manipulation and querying). Those technologies are extensible, interoperable, and platform-independent (Decker S. *et al.* 2000; Tagarelli A. *et al.* 2009), aiming to improve data modeling, annotation, manipulation, search and integration, and thus allow intelligent information retrieval on the Web (Baziz M. *et al.* 2005), which is at the core of the SW (Decker S. *et al.* 2000).

The goal of this paper is to provide a concise and comprehensive review on the technologies and tools contributing to the development and evolution of the SW, namely: knowledge bases and semantic data description, as well as XML-based data representation technologies (i.e., RDF, RDFS, OWL, and SPARQL). We also discuss the main application domains characterizing the SW, ranging over information retrieval and extraction, to lexicography and machine translation. The remainder of this paper is organized as follows. Section 2 presents the SW vision, its motivations and needs. Section 3 describes the main SW technologies. Section 4 presents the main applications and potential uses of the SW, before concluding with some future directions in Section 5.

2 SEMANTIC WEB VISION

Imagine having your favorite artist's new song downloaded automatically and added to your playlist, or imagine your phone automatically turning down the sound of all other local devices (e.g., television set, radio, laptop, etc. with a wireless volume control system) when you answer a phone call. Is that possible today, having some kind of software agent capable of understanding your needs and acting intelligently in order to fulfill them? And if there is such an agent, how could it gather all the data needed for these kinds of tasks? And most importantly, how would it understand the gathered data, analyze it and extract the bits and pieces needed for the task ahead? In the remainder of this section, we try to answer some of these questions by presenting and discussing the vision of the SW, originally introduced by the founder of the W3C, Tim Berners-Lee in 2001 (Berners-Lee T. *et al.* 2001), which promises to resolve such scenarios, and more.

2.1 Overview: What is the Semantic Web?

In order to answer the question above, we first need to clearly identify and distinguish the concepts of: *data*, *information*, *knowledge* and *metadata*. The main difference lies in the level of abstraction of each concept. *Data* is viewed as the lowest abstraction and contains no meaning whatsoever. For example, *"2001"* is considered as a number consisting of 4 digits, and highlights no information at all. For that data to be *informative*, it must be interpreted and given a well-defined meaning (such as *"the year of announcement of the Semantic Web"*) and can be therefore qualified as *information* (Zins 2007). In this context, *metadata* is viewed as a description about the data and information (such as who gave the data/information – e.g., *Wikipedia*, when was the data/information given – e.g., *published in 2002*, etc.) (Chen M. *et al.* 2009). Consequently, and at a higher level of abstraction, *knowledge* is viewed as the combination of all known data, information, and meta-data concerning a given concept or fact, as well as the semantic links between them (Spiegler 2003; Zins 2007) (like knowing that *"the year of announcement of the Semantic Web"* is *"2001"*, following *Wikipedia* in an article *published in 2002*).

In this context, we can generally distinguish between two kinds of files available on the Web: i) data documents (e.g., text files, media files, maps, graphs, etc.) designed to be accessed and understood by human users, and ii) information documents (e.g., calendars, contacts, registration info, traveling info, etc.) which can be stored and manipulated automatically by machines. The problem here resides in the separation between those two kinds of files. For example, it would be very difficult for a computer to understand the information behind a generated graph, or the meaning and contents of a picture.

This is why the original version of the Web (or the *Web 1.0*) was known as the *"Web of Documents"*, where documents are written in HTML (Hypertext Markup Language), uniquely identified by a URI (Uniform Resource Identifier) and linked between each other through hyperlinks (Bertails A. *et al.* 2010). The traditional Web then gradually grew to meet the requirements and the needs of its users allowing them to better interact with the data and information published online. Since the late 1990s, websites started gradually morphing from a means to simply access and retrieve information, into interactive platforms where users could also post and manipulate information. As a result, websites became increasingly interactive, allowing users to easily exchange ideas, discuss topics, and publish information, which soon drove the Web to another level: the *Social Web* (Web 2.0) where people are involved in publishing and also interacting with other users' published materials (Anderson P. 2007). The Web 2.0 introduced a new era of distributed data and information management with double interaction: i) horizontal: *user-user*, and ii) vertical: *user-machine*.

Based on the need for more effective user-machine interactions, Tim Berners-Lee later introduced the vision of the SW, as an extension of the current Web, in which information is associated well-defined meaning (i.e., knowledge), better enabling computers and people to interact and work in cooperation (Berners-Lee T. *et al.* 2001). In addition, endowing machines with the ability to process knowledge has extended the concept of cooperation (introduced with the Web 2.0) into a new level: *machine-machine* interaction. This kind of interaction is not based on a simple exchange of *data*, but rather on an exchange of semantically meaningful *information* well understood by machines.

2.2 Motivations: Why the Semantic Web?

The vision of the SW spurs from a set of basic needs aiming to improve Web data accessibility and performance, which we briefly describe in the following.

2.2.1 Improving Search Engines

Search engines have been developed and used for many years now on the Web, and have been improved throughout the years using many techniques, such as *interactive querying* (Derthick *et al.* 1997; Mishra *et al.* 2009) (the query is evaluated iteratively through the user relevance feedback, such as at each iteration, the user views the results and refines the query accordingly), *exploratory search* (Bozzon et al. 2010; De Virgilio et al. 2012), also known as *browsing* (navigating and exploring the information without having to formulate complex and/or explicit queries), *approximate querying* (Roussopoulos *et al.* 1995; Theobald *et al.* 2008) (answering queries with information that is close or related to what is requested), query expansion (Carpineto *et al.* 2001; Carpineto *et al.* 2012) (automatic query extension and refinement to produce better results) and *search result organization* (rearranging query results (Lin *et al.* 2003; Zhang *et al.* 2005), highlighting salient features and outliers (Kaisser *et al.* 2008; Van Leuken *et al.* 2009), to facilitate the user's task in selecting relevant answers). Yet, regardless of the technique used, search engines on

the Web generally still suffer from a lack of accuracy in retrieving search results. That is mainly because queries are processed as text-based phrases in which keywords are found and matched to the results, which sometimes generates results far from the users' intentions expressed in the queries (Baziz *et al.* 2005; Tagarelli *et al.* 2010).

In other words, given the concepts of *data* and *information* (introduced in Section 2.1), most existing systems process queries as *data* requests, neglecting most of the meaning (semantics) behind the *information* (e.g., what does the query mean? what does the user want from the query? etc,). However, evaluating the semantic relatedness between documents published on the Web is considered of key importance to improving storage and search results (Maguitman A.; Menczer F.; Roinestad H.; and Vespignani A. 2005; Tagarelli A. *et al.* 2009): grouping together similar documents, finding related documents, and given a set of documents (Baziz M. *et al.* 2005; Tagarelli A. *et al.* 2010), effectively ranking them according to their similarity (Maguitman A. *et al.* 2005). Hence there is a central need to augment existing Web search engines with semantic-based processing functionalities in order to produce more accurate results.

2.2.2 Providing New Appropriate Services

In addition to improving existing search engines, new kinds of services become more desperately needed, namely: *intelligent services*, *personalized services*, and *domain-based services*.

Intelligent services: As the Web revolves more and more around vertical interaction between men and machines (i.e., *user-machine*), the need for more sophisticated and intelligent services becomes evident, envisioning a scenario where the machine becomes more than a receiver (answering service requests), but rather an interlocutor, capable of initiating, negotiating, composing, and intelligently discovering new services (Van Den Heuvel W.J. *et al.* 2003; Lau R. 2007).

Personalized services: In addition, the Social Web experience has identified the need for a more personal engagement with the user, revolving around the user's needs, character, persona, beliefs, expectations, ergonomics and projections. For this to become real, the user needs to specify these characteristics for the machine so that it can understand them and act accordingly (Zheng Y. *et al.* 2010; Pallis G. *et al.* 2011) . For instance, a personalized geo-service could detect the locations and identify the trajectories of users, and then mine the correlation between users and locations, allowing to i) connect users that share similar travel trajectories, ii) provide users with generic travel recommendations (e.g., most interesting locations), and iii) personalize friend and location recommendations (Zheng Y. *et al.* 2010).

Domain-based services: Geo-services (Zheng Y. *et al.* 2010) are also a good example of domain-based services, related to geographic information systems (*GIS*). In this context, knowledge specific to the domain at hand is usually shared between agents and services acting in the domain. For example, in the GIS domain, concepts such as "*GPS*" (Global Positioning System) or "*GML*" (Geographic Markup Language) are often used, and often designate (each) the same meanings. Hence the need for domain-based services, built among domain-specific knowledge representations (e.g., domain-specific dictionaries, taxonomies, or ontologies), in order to reduce the redundancy factor, improve service accuracy, and help speed up service processing (Rocco D. *et al.* 2005; Zhang H. *et al.* 2010). In other words, this requires a common framework for organizing knowledge in a specific domain, which is both accurate and complete in describing the semantics of the information at hand (Li H. *et al.* 2007).

2.2.3 Improving Data Accessibility

One of the main problems with data published on the Web at the moment is that it is not in a form that can be effectively and easily used (Berners-Lee T. *et al.* 2001; Berners-Lee T. *et al.* 2009). Data can be stored in different ways (spreadsheets, databases, etc.) and is not usually posted on the Web in its original form. Rather, certain bits of information deduced from the data itself are published online (following certain constraints, e.g., target audience, online storage space, etc.), and presented often in a more user-friendly format, such as plain text, graphs, charts, tables, etc. Hence, online information is often specific (to a certain audience) and might not be reusable by different users and/or applications (Ding L. *et al.* 2005; Berners-Lee T. *et al.* 2009). For example, when searching for a map on the Web, one user might be interested in street names, while another user searches for restaurants. Hence, the corresponding online geographic data has to be complete, endowed with flexible and efficient data access services providing each user with the information which best answers her needs (Ding *et al.* 2005; Hastings *et al.* 2006).

2.2.4 Better Data Integration & Presentation

As mentioned earlier, the Web is a sum of contents linked together via hyperlinks. These hyperlinks reflect (in one way or another) certain semantic relationships between documents (Brin S. and Page L. 1998; Kleinberg J. 1999), which have been proven very effective in answering queries and identifying relevant Web pages (Brin S. and Page L. 1998). In fact, sophisticated algorithms such as PageRank (Brin S. and Page L. 1998) and HITS (Kleinberg J. 1999) have been developed to analyze link structures in order to rank Web pages. However, most documents published on the Web remain flat, i.e., consisting of unstructured data, which limits the performance of existing (conventional) search engines.

For example, a traveler seeking a train ticket to get to the airport would have to access two separate Web pages and cross-match the data to find the best tickets available. Yet, if the contents of Web pages were structured in a way to access specific information related to departure/arrival dates (rather than accessing the flat pages as a whole), then the data would be crisscrossed automatically, providing the user with the bundled data (e.g., a combination of a train and a plane ticket) in a single view.

Hence, the need to structure data published on the Web becomes critical, where not only documents as a whole, but rather structured information within the documents are linked together. This is more recently known as the concept "Linked data" (Heath T. *et al.* 2011; Bizer C. *et al.* 2012), and is at the core of the SW vision, especially with the dawn of XML in the late 1990s, as a standard (semi-)structured data representation and exchange model on the Web (Section 3).

3 SEMANTIC WEB TECHNOLOGIES

3.1 Overall Architecture

The SW vision is based on three main concepts: *objects*, *labels* and *links*. Objects designate any piece of data posted on the Web, e.g., Web pages, services, media files, text, etc. Each object is usually described with metadata, known as *data labeling* (augmenting data with descriptive labels), and is uniquely identified by an URI allowing the object to be unambiguously linked with other objects. Here, some golden rules need to be followed while realizing the SW architecture (Bertails A. *et al.* 2010):

- Every object is labeled,
- Labels are readable by software agents and humans,
- Labels describe corresponding objects accurately,
- Labels are located in a common readable environment for software agents and humans to explore, making objects accessible as resources.

In this context, a hierarchy of technologies, mainly: XML, RDF, RDFS, OWL, and SPARQL, were gradually normalized by the W3C in the last decade aiming to fulfill the SW architecture (also known as the "*The Semantic Web Stack*", Figure 1). Three layers can be distinguished in the stack: i) the naming and addressing layer, ii) the syntax layer and iii) the semantic layers. The naming and addressing layer associates an object with a unique identifier, i.e., a URI (Universal Resource Identifier) or an IRI (Internationalized Resource Identifier) for multilingual Web addresses (Ishida R. 2008). The syntax layer structures the data in a tree-like structure, using XML-based constructs and namespaces. Finally, to append semantic meaning to data, a semantic layer is added which associates labels to data objects, involving all other technologies, from ontology to query and rule-based languages (Joo J. 2011).

In the remainder of this section, we briefly present each of the main technologies making up the building blocks of the SW architecture, namely: knowledge bases, XML, RDF, RDFS, OWL, and SPAQRL.

3.2 Description Logics & Knowledge Bases

Description Logics (DLs) are a family of languages for Knowledge Representation (KR) and Knowledge Inference (KI) (Champavère J. 2010). On one hand, KR in Artificial Intelligence (AI) underlines a means to represent and describe *knowledge*, to be stored in so-called Knowledge Bases (KBs), i.e., repositories of machine-readable *knowledge*, available for automated processes (software agents) to use and exploit, hoping to achieve intelligent processing capabilities. On the other hand, KI is the knowledge deduced by an inference engine, working within or alongside the automated process, based on a predefined KB. As for DLs, many languages have been proposed such as: Propositional Logic, First-Order Logic, Temporal Logic, Fuzzy Logic, etc. each of which with its set of properties and application(s), and have been exploited, in one way or another, in semantic data analysis (Heflin J. 2000; Terzi E. *et al.* 2003).

In this context, every KB system based on DL is composed of a Terminology-Box (T-Box) and an Assertion-Box (A-Box). The *T-Box* underlines the set of concept definitions, while the *A-Box* consists of the collection of concept instances (also called individuals). In comparison with a relational database, the *T-Box* is similar to the structure of the tables (database schema) whereas the *A-Box* is more like the data rows (tuples) inserted into the tables (Bertails *et al.* 2010; Champavère 2010). Here, KR structures such as taxonomies, thesauri, and ontologies, etc. have been investigated and developed (in the domains of natural language processing and information retrieval), in order to define, organize and link concepts in a KB (Jiang and Conrath 1997).

A KB usually comes down to a *semantic network* which is basically a graph consisting of nodes and arcs, organizing words/expressions in a semantic space (Richardson R. and Smeaton A. 1995; Jiang J. and Conrath D. 1997) (Figure 2). Each node represents a concept underlining a group of words/expressions (or URLs such as with ODP – Open Directory Project (Maguitman A. *et al.* 2005)). Arcs underline the semantic links connecting the concepts, representing semantic relations (synonymy, hyponymy, meronymy, etc. (Miller G. 1990; Richardson R. and Smeaton A. 1995)). Typical examples of lexical KBs are Roget's thesaurus (Yaworsky D. 1992) and WordNet (Miller G. 1990) (cf. Figure 2).

In such structures, knowledge is usually processed as sets of triplets: *concept1-relationship-concept2*, or as more commonly known: *subject-predicate-object* triplets (Guo Y. *et al.* 2007; Champavère J. 2010). This corresponds to the triplet-based representation: *objects*, *labels*, *links*, emphasized in the SW (Section 3.1).

3.3 XML & Interoperability

The distributed nature of the Web, as a decentralized system running over multiple platforms and exchanging information between multiple heterogeneous sources, has underlined the need to manage *semantic interoperability*, i.e., the ability to automatically interpret *information* in Web documents exchanged between different sources, in a semantically meaningful way in order to produce useful results for efficient information management and search applications (Fuhr and Großjohann 2001; Guo *et al.* 2007).

In this context, XML was introduced as a data representation model that simplifies the tasks of interoperation and integration among heterogeneous data

sources (Bray *et al.* 2008). It allows to represent data in a (semi-) structured document, consisting of hierarchically nested information, made of a set of atomic and complex elements (i.e., containing sub-elements) as well as atomic attributes, thus incorporating structure and content in one entity (cf. Figure 3). Each tag is either an empty-element with attribute-value pairs, or an element with contents between its start-tag and end-tag. In contrast with HTML designed for visual markup, XML tagging concerns data contents, and is not limited to a fixed vocabulary, but rather allows flexible and extensible application-based vocabularies expressed using dedicated grammar definitions (such as DTD – Document Type Definition (Bray *et al.* 2008) or XSD – XML Schema Definition (Gao *et al.* 2009)), specifying allowable combinations and nesting of tag names, attribute names, and the rules they adhere to in the documents.

Making use of XML to index, represent, retrieve and compare complex objects has been proven successful, in multimedia applications (e.g., SVG, SMIL, X3D and MPEG-7 are only some examples of XML-based multimedia data and meta-data representations), in scientific data description formats (e.g., XSIL and XDF for general scientific data, CML for chemical molecule descriptions, BIOML for protein and gene sequence descriptions, etc.), and in geographic data representations (namely GML and KML for describing geo-referenced entities). XML also provides a common data serialization and exchange format between different programming languages (e.g., php, jsp, asp, java, C#, etc.).

To sum up, XML was shown most effective in exchanging data (*data interoperability*), yet has been proven limited when it comes to handling semantics (*semantic interoperability*). E.g., an object such as "*person*" in Figure 3, with properties: "*first name*", "*last name*", etc. can be serialized in different ways in XML. While semantically identical, these serializations are treated differently by different XML engines, since XML only specifies the syntactic and structural features of the data without any further semantic meaning (which is where RDF comes to play).

3.4 RDF & Semantics

While XML addresses the syntactic and structural properties of data, RDF (Resource Description Framework) (Manola *et al.* 2004) builds on XML to better manage *semantic interoperation*. RDF is an XML-based data model designed to standardize the definition and use of metadata, in order to better describe and handle data semantics. RDF was designed to meet the following goals:

- Having a simple data model,
- Having formal semantics and provable inference,
- Using an extensible URI-based vocabulary,
- Supporting the use of XSD data-types,
- Transparent description of Web resources.

The RDF data representation model is based on triplets (*Object*, *Attribute*, *Value*), more commonly known as $A(O,V)$. A triplet binds an attribute value to an object, giving the relationship a semantic meaning. Objects, attributes and values underline any kind of Web resources, identified using URIs. Values can also contain literal (text) contents. For example, consider three resources: *person*, *name* and *marriage*, and instantiate them as follows: *person*: p_1, p_2, *name*: n_1, n_2 and *marriage*: m_1. Basic triplets that can be modeled here are: *name*(p_1, n_1), *name*(p_2, n_2) and *marriage*(p_1, p_2). But in this type of representation, some of the semantic is lost. In fact, following the latter triplet, the *marriage* attribute joins two people p_1 and p_2, indicating that p_2 is the *marriage* of p_1 (expressed in XML format in Figure 4), which does not sound semantically precise, especially when automatically interpreted by machines. What we would rather have is a more (semantically) specific expression indicating that p_2 *is married to* p_2.

To do so, RDF allows the creation of so-called predicates or (semantic) properties, which underline (more specific) subsets of the resources. For instance, to link the concept *person* to the concept *name*, a property *hasName* can be used, such as *hasName*(p_1, n_1) precisely indicates – even for a machine - that p_1 is a *person* and n_1 is her/his *name*. Also, a property *IsMarriedTo* (sub-set of *marriage*) can be used to link p_1 and p_2: *IsMarriedTo*(p_1, p_2). In the SW, the triplet is no longer called: *object-attribute-value*, but rather by *resource-property-value* (Decker *et al.* 2000).

Properties underline the main advantage and added value of RDF over XML (Decker *et al.* 2000), as a more suitable technology for semantic interoperability and portability, since they allow RDF to exchange, share and reuse (semantic) information between applications. Note that the RDF has a formal semantics (Hayes P. 2004), with a predefined namespace (i.e., *rdf*) and elements prefix tags (such as *rdf:type*, *rdf:Property*, *rdf:XMLLiteral*, etc.) which provides a solid basis for creating and reasoning about the meaning of an RDF expression. In particular, it supports rigorously defined notions of semantic relations and dependencies which provide a basis for defining rules of inference in the RDF model (Klyne *et al.* 2004).

Nonetheless, it is important to note that RDF by itself is just a data model, i.e., an *A-Box*. What really gives away the intended semantics behind this data model is the use of a solid, rigorous and well defined vocabulary, i.e., a *T-Box*, which is where RDFS (RDF Schema) comes to play.

3.5 RDF Schema (RDFS)

RDFS (Brickley and Guha 2004) is the equivalent of an XML grammar (DTD or XSD) for an RDF document. It is like a *T-Box* for RDF (the schema of a KR, cf. Section 3.2) shaping the model in which the RDF data instances will be inserted. RDFS models and manipulates classes, similarly to an object-oriented programming language. The main difference lies in the definition of classes and properties: instead of defining a class in terms of the properties its instances may have, RDFS describes properties in terms of the classes of resources to which they apply (Brickley and Guha 2004) (as briefly explained with the *HasName* and *IsMarriedTo* examples in Section 3.4, where the properties were defined in terms of their resources).

In other words, RDFS is a semantic extension of RDF (Manola *et al.* 2004) (with a dedicated namespace: *rdfs*), providing mechanisms for describing groups of related resources and the semantic relationships between these resources (Brickley D. and Guha R. V. 2004), especially at the object-oriented level with hierarchy and heritage implementations (using constructs such as \rdfs:Class, *rdfs:SubClassOf*, *srds:SubPropertyOf* e.g.), as well as at the predicate's level specifying the property's domain and range of application (using *rdfs:domain* and *rdfs:range*). This means that if an individual is created without being typed (i.e., it does belong to a defined class) and is linked to another resource by a property, the inference engine would be able to class this individual (using the property's domain and/or range) to the corresponding resource type using the schema of the property in question. In short, an effective semantic inference engine requires both the RDF data instances (the A-Box) and their corresponding RDF Schemas (the T-Box) in order to run properly (Hayes P. 2004; Klyne *et al.* 2004).

Yet, despite its expressiveness, RDFS carries some limitations (Antoniou G. *et al.* 2004):

- Local scope of properties: it does not allow restrictions or generalizations of properties,
- Disjointness of classes: two classes cannot be formally identified as disjoint,
- Boolean combinations of classes: it does not allow Boolean set operators (e.g., union, intersection, complement, etc.) when creating classes,
- Cardinality restrictions: it is not possible to define a restriction on how many distinct values a property may or must take,
- Special characteristics of properties: it does not allow transitive, unique and/or inverse properties.

Hence, while RDFS is semantically more expressive than RDF in describing basic Web resources (e.g., Web pages), it still lacks in expressiveness (as shown above) especially when describing more complex resources such as taxonomies or ontologies (Antoniou G. *et al.* 2004; Hayes P. 2004), which is where OWL comes to play.

3.6 Web Ontology Language (OWL)

While basic Web resources can be effectively described using binary ground predicates (using RDF) and/or subclass and property hierarchies (using RDFS), yet Web experts have identified the need for more semantic expressiveness: building common semantic reference information sources, or so-called Web Ontologies, serving as knowledge references for software agents when automatically processing Web resources (similarly to dictionaries and encyclopedias serving as knowledge references for human agents) (Dean M. *et al.* 2004; Garcia-Castro R. *et al.* 2010).

Hence, filling the gaps of RDF and RDFS, W3C has introduced OWL (Ontology Web Language) as a standard for describing ontologies on the Web. OWL it is built upon RDF and RDFS, and inherits its most basic elements from RDFS (including constructs such as *owl:Class*, *owl:ObjectProperty*, *owl:DataTypeProperty* which extend the expressiveness of their *rdfs:Class* and *rdfs:Property* ancestors) allowing property specialization, the identification of disjoint classes, specifying cardinality or data-type restrictions etc. (Antoniou G. *et al.* 2004).

Due to the semantic richness of OWL constructs which might be complex to use in everyday scenarios, OWL has been presented in three W3C specifications (Dean M. *et al.* 2004; Garcia-Castro R. *et al.* 2010; McGuinness D.L. *et al.* 2012), depending on the reasoning level: i) OWL Lite, ii) OWL-DL and iii) OWL full.

OWL Full is the basic version using all the OWL language primitives (as briefly describe above). It is fully compatible with RDF(S), both syntactically and semantically, but can be complex to handle given its powerful expressiveness.

OWL-DL (OWL-Description Logic) is a sub-language of *OWL Full* which restricts the way the constructs from OWL and RDF(S) can be used (Heflin J. 2007; Garcia-Castro R. *et al.* 2010). For instance, OWL-DL requires that every resource either be a class, object property, data-type property or instance, but cannot be treated as both a class and an instance at the same time. Furthermore, the category of each resource must be explicit in all ontologies (i.e., each resource must have an *rdf:type* statement). In other words, one cannot use a resource as a *class* without explicitly describing it as such. Note that such restrictions do not exist in *OWL Full* (where it is possible to treat a class as an instance, and there is no need to explicitly declare the type of each resource). Also note that *OWL-DL* is not compatible with RDF but is more efficient in reasoning support as an expressive Description Logic (DL) (Champavère J. 2010).

OWL Lite is a sublanguage of *OWL-DL* which excludes enumerated classes, disjointness statements and arbitrary cardinality (among others). It is intended for users with simple modeling needs and is user and implementation friendly (Garcia-Castro R. *et al.* 2010).

Having defined powerful languages and constructs (RDF, RDFS, and OWL) to describe elaborate semantic information, the W3C has identified the need for finding a powerful means to accessing and querying them (Garcia-Castro R. *et al.* 2010; Joo J. 2011), which is where SPARQL comes to play.

3.7 Data Manipulation & Querying

For any database to be useful (other than for data storage), it needs to be queryable. In other words, the value of (semantic information) contents depends on how easy it is to search, access and manage (Pereira F. 2006). Hence, several solutions have been proposed for querying XML, RDF and OWL instances, namely XQuery (Chamberlin D. *et al.* 2001) for XML-based (and RDF/XML) documents, and SPARQL (Prudhommeaux E. *et al.* 2008) for RDF-based (and OWL) documents. These languages are specially designed to resemble SQL in their grammars and constructs, to facilitate their usage by programmers.

For instance, a simple query in SPARQL would have the following form: *SELECT * WHERE* {*S*, *P*, *O*} Where (*S*, *P*, *O*) are the RDF triplets (subject *S*, predicate *P*, object *O*). More complex queries can be created in SPARQL to search about anything in an RDF document (Hartig O. *et al.* 2009). By adding other namespaces such as RDFS or OWL, SPARQL can be extended to query also triples from an ontology document.

Yet syntactically speaking, XQuery and SPARQL are not easy-to-use (straightforward) query languages, and generally require deep knowledge and special skills in XML and RDF-based languages to be manipulated efficiently. Recent research efforts have focused on developing simplified tools or alternatives, including visual interfaces (Ambrus O. *et al.* 2010), keyword-based querying (Shekarpour S. *et al.* 2011), and eventually programming APIs (Jena (Apache Jena), OWL API (Protege-OWL API), etc.) to help create, manipulate and query structured, semantically rich and ontology-based documents. In addition, XQyery and SPARQL are based on exact matching and do not support ranked queries via textual/structural similarity. Hence, several attempts have been made to extend these query languages in order to support ranked results (Amer-Yahia S. *et al.* 2004; Marian A. *et al.* 2005; Theobald M. *et al.* 2008).

4 SEMANTIC WEB APPLICATIONS

The use of SW technologies is central in wide spectrum of applications, ranging over: information retrieval, information extraction, machine translation, content analysis, and lexicography.

Information Retrieval: As mentioned earlier, state-of-the-art search engines do not use explicit semantics to prune out documents which are not relevant to a user query (Navigli R. and Velardi P. 2003; Baziz M. *et al.* 2005). Hence, semantic document and query indexing (i.e., associating accurate semantic labels to document/ query concepts, with respect to a reference KB) would allow it to eliminate documents containing the same words used with different meanings (thus increasing precision) and to retrieve documents expressing the same meaning with different wordings (thus increasing recall), e.g. (Greenberg J. 2001; Navigli R. and Velardi P. 2003; Baziz M. *et al.* 2005; Schenkel R. *et al.* 2005; Li Y. *et al.* 2006; Theobald M. *et al.* 2008).

Information Extraction: Extracting semantically related concepts form a corpus (also called semantic categories or labels), which is particularly useful for part-of-speech tagging (i.e., the assignment of parts of speech to target words, i.e., concepts, in context) (Allan J. and H. Raghavan 2002), named entity resolution (i.e., the classification of target textual items into predefined categories, i.e., semantic concepts) (Pilz A. *et al.* 2011), and text categorization (i.e., the assignment of predefined labels, i.e., concepts, to target texts) (Bloehdorn S. *et al.* 2004; Bawakid A. *et al.* 2010).

Machine Translation: The automatic identification of the correct translation of a word in context (called *machine translation*) is a critical task in the SW vision, as it requires word sense disambiguation (i.e., associating right sense for the right word, among a set of possible senses given a reference KB) (Ide N. and Veronis J. 1998; Mihalcea R. 2006), given that the disambiguation of texts should help translation systems choose better candidates. Machine translation becomes also central for live speech translation techniques (Vickrey D. *et al.* 2005; Carpuat M. *et al.* 2007; Chan Y.S. *et al.* 2007).

Content analysis: It underlines the analysis of the general content of a text in terms of its ideas, themes, etc. which is gaining importance in various applications such as i) blog classification (e.g., introducing simple and effective methods to semantically classify blogs, determining their main topics, and identifying their semantic connections (Berendt B. and Navigli R. 2006; Berendt B. *et al.* 2009), and ii) semantic social network analysis (e.g., disambiguation of entities in social networks, and identifying semantic relations between users based on their published materials (Aleman-Meza B. *et al.* 2008; Erétéo G. *et al.* 2009; Passant A. *et al.* 2012)).

Lexicography: It underlines the creation of dictionaries or ontologies (i.e., semantic references). While lexicography was restricted to human experts, with the advent of the SW, there has been a growing interest in the field of automatic ontology generation, using empirical sense groupings and data analysis (statistically significant indicators of context for new or existing senses (Cimiano P. 2006; Kilgarriff A. 2006; Zhao L. *et al.* 2011)), and the integration/combination of existing semantic references and structured documents to produce a new one (e.g., creating domain or application-specific ontologies, etc. (Navigli R. *et al.* 2004; Ye S. *et al.* 2007; Umer Q. *et al.* 2012)).

To sum up, the SW vision can potentially benefit from all of the above-mentioned applications, as it inherently needs domain-oriented and unrestricted sense disambiguation to deal with the semantics of Web documents, and enable interoperability between systems, ontologies, and users.

3 CONLCUSIONS & TRENDS

In this paper, we gave a brief overview on the SW vision and underlying technologies, ranging from knowledge bases and semantic representation, to extensible and interoperable XML-based data representation technologies (namely RDF, RDFS, OWL and SPARQL), aiming to improve Web data accessibility and performance through data modeling, annotation, manipulation, search and integration, on the Web. We also discussed some the main services and applications promoted by the SW, ranging from intelligent information retrieval, to information extraction, machine translation, content analysis and lexicography.

Nonetheless, realising the SW vision still faces many challenges under investigation:

- *Creating comprehensive ontologies* containing all the concepts required in a given domain remains an extremely difficult task, namely owing to the difficulty of managing a huge KB (e.g., computing the semantic similarity between two concepts using the WordNet taxonomy requires several hours (Maguitman 2005; Tekli 2011). This problem could be solved by creating several domain-based ontologies, which is currently a hot research topic (Navigli 2004; Ye S. 2007; Umer Q. *et al.* 2012).
- *Compromising between expressiveness and reasoning* is a very delicate issue: The more the language is expressive the harder the reasoning to be achieved (Heflin J. 2000). Hence, choosing the DL to be used, as well as the level of semantic details (in the KBs) remains a very critical task.
- Since ontologies are created separately by different developers, some concepts are being redefined constantly. Hence, *mapping structurally and semantically-rich documents* (XML and/or RDF-based) (Araújo S. *et al.* 2011; Tekli J. *et al.* 2012b; Tekli J. *et al.* 2012a; Tzitzikas Y. *et al.* 2012) and ontologies (Ehrig M. and Sure Y. 2004; Ming M. *et al.* 2008; Shvaiko P. and Euzenat J. 2008) is central to help lower redundancy and increase efficiency.
- *Simplifying the use and manipulation of KBs* and semantic references, including visual interfaces (Ambrus O. *et al.* 2010), approximate querying, (Theobald M. *et al.* 2008), keyword-based querying (Shekarpour S. *et al.* 2011), and eventually programming APIs (Jena, OWL API, etc.) to help create, manipulate and query semantically rich and ontology-based documents. Investigating Visual mashups is also promising in this field (Grammel *et al.* 2010; Liu D. *et al.* 2011; Tekli G. *et al.* 2011).
- Fostering *service intelligence* and *atomicity* (the ability of services to work and interact automatically, central in *user-machine* and *machine-machine* interactions), remains one of the most upcoming challenges of the SW. Allowing software agents to perform intelligent tasks relies on the awareness (intelligence) of the software agents, and their ability to learn, act and evolve with time. Here, *semantics* and *linked-data* are currently heavily investigated as the backbone of intelligent software agents (Van Den Heuvel W.J. *et al.* 2003; Lau R. 2007).
- *Improving privacy protection* strategies to reduce the information disclosure caused by data sharing and linkage. It is in essence worthy to note that data sharing and linkage are not always beneficial and could be dangerous in several situations (social networks, health, etc.). New relevant privacy protection solutions must be provided in order to protect sensitive information (which might be multimedia-based) especially those that Web 3.0 users would like to keep private (Gabillon A. *et al.* 2010; Al Bouna B. *et al.* 2012).

We hope that our presentation of the SW vision, technologies and applications will help strengthen further research on the subject.

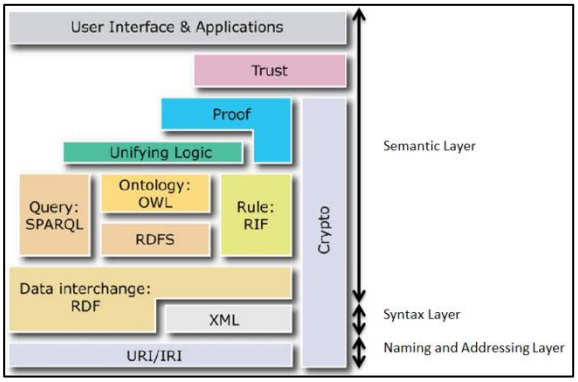

Figure 1. The Semantic Web Stack

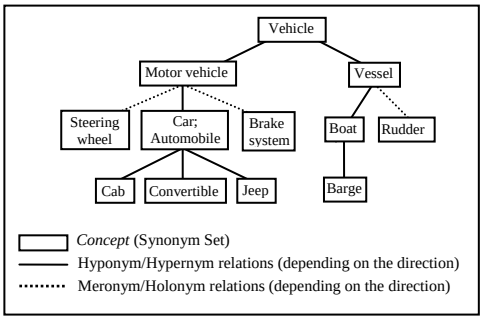

Figure 2: An extract from the WordNet semantic network.

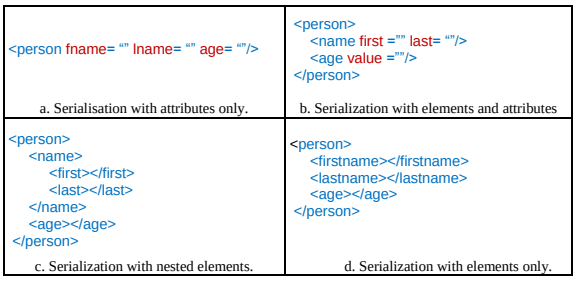

Figure 3: Different XML serializations of an entity *person.*

```
<marriage id= "m1">
   <person id= "p1" name= "n1"/>
   <person id= "p2" name= "n2"/>
</marriage>
```

Figure 4: XML representation of a *marriage* relationship.

REFERENCES

Al Bouna B., *et al.* (2012). *A Fine-Grained Image Access Control Model*. SITIS'12 pp. 603-612.

Aleman-Meza B., *et al.* (2008). *Scalable Semantic Analytics on Social Networks for Addressing the Problem of Conflict of Interest Detection*. ACM TWeb 2(1):7.

Allan J. and H. Raghavan (2002). *Using Part-of-Speech Patterns to Reduce Query Ambiguity*. ACM SIGIR, pp. 307-314, Tampere.

Ambrus O., *et al.* (2010). *Konduit VQB: a Visual Query Builder for SPARQL on the Social Semantic Desktop*. VISSW'10.

Amer-Yahia S., *et al.* (2004). *FleXPath: Flexible Structure and Full-Text Querying for XML*. Proceedings of the ACM International Conference on Management of Data (SIGMOD) pp. 83-94.

Anderson P. (2007). *What is Web 2.0? Ideas, technologies and implications for education.* JISC Tech. and Standards Watch, 64.

Antoniou G., *et al.* (2004). *Web Ontology Language: OWL*. Handbook on Ontologies pp. 67-92.

Apache Jena. (January 2012). *ARQ - A SPARQL Processor for Jena*. http://jena.apache.org/documentation/query/index.html (Jan 2012.

Araújo S., *et al.* (2011). *SERIMI - Resource Description Similarity, RDF Instance Matching and Interlinking*. International Semantic Web Conference (ISWC'11).

Bawakid A., *et al.* (2010). *A Semantic-based Text Classification System*. IEEE 9th International Conference on Cybernetic Intelligent Systems (CIS'10), pp. 1-6.

Baziz M., *et al.* (2005). *A concept-based approach for indexing documents in IR.* INFORSID 2005 pp. 489-504, Grenoble.

Berendt B., *et al.* (2009). *Semantics-based Analysis and Navigation of Heterogeneous Text Corpora: The Porpoise News and Blogs Engine*. Web Mining Applications in E-commerce and E-services Studies in Computational Intelligence 172:45-64.

Berendt B. and Navigli R. (2006). *Finding Your Way Through Blogspace: Using Semantics for Cross-Domain Blog Analysis*. AAAI Spring Symposium (AAAIS) on Computational Approaches to Analysing Weblogs pp. 1–8.

Berners-Lee T., *et al.* (2001). *The Semantic Web*. Scientific American 284(5):1:19.

Berners-Lee T., *et al.* (2009). *Tim Berners-Lee Looks Back: the "//" in Web Addresses Was Unnecessary*. http://tinyurl.com/bgse2cw.

Bertails A., *et al.* (2010). *The Semantic Web: The Internet and Tomorow's Web*. Industrial Realities (in French) pp. 80-89.

Bizer C., *et al.* (2012). *WWW'12 Workshop on Linked Data on the Web*. CEUR Workshop Proceedings 937, CEUR-WS.org.

Bloehdorn S., *et al.* (2004). *Text Classification by Boosting Weak Learners based on Terms and Concepts*. ICDM pp. 331-334.

Bozzon A., *et al.* (2010). *Liquid Query: Multi-Domain Exploratory Search on the Web*. Proceedings of the 19th International Conference on World Wide Web (WWW '10), pp. 161-170, NY.

Bray T., *et al.* (2008). *Extensible Mark-up Language (XML) 1.0 - 5th Edition. W3C recommendation, 26 November 2008*. Retrieved November 2008, from http://www.w3.org/TR/REC-xml/.

Brickley D. and Guha R. V. (2004). *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation http://www.w3.org/TR/rdf-schema/.

Brin S. and Page L. (1998). *The Anatomy of a Large Scale Hypertextual Web Search Engine*. In Computer Networks and ISDN Systems 30 (1-7):107-117.

Carpineto C., *et al.* (2001). *An Information-Theoretic Approach to Automatic Query Expansion*. ACM TOIS, 19(1):1-27.

Carpineto C., *et al.* (2012). *A Survey of Automatic Query Expansion in Information Retrieval,* . ACM Computing Survey, 44(1):1.

Carpuat M., *et al.* (2007). *Improving Statistical Machine Translation using Word Sense Disambiguation*. EMNLP-CoNLL'0, pp. 61-72.

Chamberlin D., *et al.* (2001). *XQuery : A Query Language for XML, http://www.w3.org/TR/2001/WD-xquery-20010215*. (May 2009).

Champavère J. (2010). *From Knowledge Representation to the Semantic Web: An Overview*. pp.14 (in French).

Chan Y.S., *et al.* (2007). *Exploiting Parallel Texts for Word Sense Disambiguation in the English all-Words Tasks*. Proceedings of SemEval'07, pp. 253–256, Prague, Czech Republic.

Chen M., *et al.* (2009). *Data, Information and Knowledge in Visualization*. IEEE CGA, 29(1):12-19.

Cimiano P. (2006). *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications.* Springer, NY.

De Virgilio R. et al. (2012). *Semantic Search over the Web*. Data-Centric Systems and Applications, 10.1007/978-3-642-25008-8 3.

Dean M., *et al.* (2004). *OWL Web Ontology Language Reference*. W3C Recommendation, http://www.w3.org/TR/owl-ref/.

Decker S., *et al.* (2000). *The Semantic Web: The Roles of XML and RDF*. IEEE Internet Computing 4(5):63-74.

Derthick M., *et al.* (1997). *An Interactive Visual Query Environment for Exploring Data*. ACM Symposium on User Interface Software and Technology (UIST '97), ACM Press pp. 189-198.

Ding L., *et al.* (2005). *Boosting Semantic Web Data Access using Swoogle*. Proceedings of AAAI'05, (4): pp. 1604-1605

Ehrig M. and Sure Y. (2004). *Ontology Mapping - an Integrated Approach*. Proceedings of the European Semantic Web Conference (ESWC) pp. 76-91. Heraklion, Greece.

Erétéo G., *et al.* (2009). *Semantic Social Network Analysis*. CoRR abs/0904.3701.

Fuhr N. and Großjohann K. (2001). *XIRQL: A Query Language for Information Retrieval*. ACM-SIGIR, pp. 172-180. New Orleans.

Gabillon A., *et al.* (2010). *A View Based Access Control Model for SPARQL*. NSS'10 pp. 105-112.

Gao S., *et al.* (2009). *W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures*. W3C recommendation, http://www.w3.org/TR/xmlschema11-1/ Retrieved May 2011.

Garcia-Castro R., *et al.* (2010). *Interoperability Results for Semantic Web Technologies using OWL as the Interchange Language*. Web Semantics, 8(4):278-291.

Grammel L., *et al.* (2010). *A Survey of Mashup Development Environments.* The Smart Internet 137-151.

Greenberg J. (2001). *Automatic Query Expansion via Lexical-Semantic Relationships*. Journal of the American Society for Information Science, 52(5):402–415.

Guo Y., *et al.* (2007). *A Requirements Driven Framework for Benchmarking Semantic Web Knowledge Base Systems*. IEEE Trans. Knowl. & Data Eng. 19(2): 297-309.

Hartig O., *et al.* (2009). *Executing SPARQL Queries over the Web of Linked Data*. ISWC'09, pp. 293-309.

Hastings S., *et al.* (2006). *Web Services Data Access and Integration*. The XML Realization, (WS-DAIX) Specification, Version 1.0.

Hayes P. (2004). *RDF Semantics*. W3C Recommendation, http://www.w3.org/TR/rdf-mt/.

Heath T., *et al.* (2011). *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool.

Heflin J. (2000). *Knowledge Representation on the Internet: Achieving Interoperability in a Dynamic, Distributed Environment*. PhD Thesis, University of Maryland, USA.

Heflin J. (2007). *An Introduction to the OWL Web Ontology Language*. http://www.cse.lehigh.edu/~heflin/IntroToOWL.pdf.

Ide N. and Veronis J. (1998). *Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art*. Computational Linguistics 24(1):1-40.

Ishida R. (2008). *An Introduction to Multilingual Web Addresses*. International/articles/idn-and-iri/, http://www.w3.org/.

Jiang J. and Conrath D. (1997). *Semantic Similarity based on Corpus Statistics and Lexical Taxonomy*. Proceedings of the International Conference on Research in Computational Linguistics.

Joo J. (2011). *Adoption of Semantic Web from the perspective of technology innovation: A grounded theory approach.* Inter. Journal of Human-Computer Studies. 69(3):139-154.

Kaisser M., *et al.* (2008). *Improving Search Results Quality by Customizing Summary Lengths*. ACL'08 pp.701-709.

Kilgarriff A. (2006). *Word Senses*. Word Sense Disambiguation: Algorithms and Applications pp.29–46, Springer, New York, NY.

Kleinberg J. (1999). *Authoritative Sources in a Hyperlinked Environment*. Journal of ACM 46(5):604–632.

Klyne G., *et al.* (2004). *Resource Description Framework (RDF): Concepts and Abstract Syntax.* . W3C Recommendation REC-rdf-concepts-20040210 http://www.w3.org/TR/rdf-concepts/.

Lau R. (2007). *Towards a Web Services and Intelligent Agents-based Negotiation System for B2B ECommerce*. Electronic Commerce Research and Applications archive 6(3):260-273

Li H., *et al.* (2007). *Towards Semantic Web Services Discovery with QoS Support using Specific Ontologies* SKG'07, pp. 358 - 36.

Li Y., *et al.* (2006). *Term Disambiguation in Natural Language Query for XML*. FQAS, LNAI 4027, pp. 133–146.

Lin W.H., *et al.* (2003). *Web Image Retrieval Re-Ranking with Relevance Model*. IEEE WIC'03, pp. 242-249.

Liu D., *et al.* (2011). *An Approach to Construct Dynamic Service Mashups Using Lightweight Semantics*. ICWE'11, pp. 13-24.

Maguitman A., *et al.* (2005). *Algorithmic Detection of Semantic Similarity*. WWW Conference, pp. 107-116.

Maguitman A.; Menczer F.; Roinestad H.; and Vespignani A. (2005). *Algorithmic Detection of Semantic Similarity*. WWW, 107-116.

Manola F., *et al.* (2004). *Resource Description Framework (RDF) Primer : Model and Syntax Specification*. W3C Recommendation http://www.w3.org/TR/rdf-primer/.

Marian A., *et al.* (2005). *Adaptive Processing of Top-k Queries in XML*. ICDE Conference, pp. 162-173.

McGuinness D.L., *et al.* (2012). *OWL 2 Web - Ontology Language Document Overview*. W3C Proposed Edited Recommendation http://www.w3.org/TR/owl2-overview/.

Mihalcea R. (2006). *Knowledge-based Methods for WSD*. In Word Sense Disambiguation: Algorithms and Applications, E. Agirre and P. Edmonds pp. 107–131, Springer, New York.

Miller G. (1990). *WordNet: An On-Line Lexical Database*. International Journal of Lexicography 3(4).

Ming M., *et al.* (2008). *A Harmony Based Adaptive Ontology Mapping Approach*. SWWS'08, pp. 336-342.

Mishra C *et al.* (2009) *Interactive Query Refinement*, EDBT, 862-873.

Navigli R., *et al.* (2004). *Learning Domain Ontologies from Document Warehouses and Dedicated Websites*. Computational Linguistics 30(2):151–179.

Navigli R. and Velardi P. (2003). *An Analysis of Ontology-based Query Expansion Strategies*. In Proceedings of IJCAI'03.

Pallis G., *et al.* (2011). *Online Social Networks: Status and Trends*. New Directions in Web Data Management (1): 213-234.

Passant A., *et al.* (2012). *Special issue on real-time and ubiquitous social semantics*. Semantic Web 3(2):113.

Pereira F. (2006). *Technologies for Digital Multimedia Communications: An Evolution Analysis of MPEG Standards*. China Communications Journal.

Pilz A., *et al.* (2011). *From names to entities using thematic context distance*. CIKM 2011 pp. 857-866.

Protege-OWL API Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine. http://protege.stanford.edu/plugins/owl/api/ (January 2012).

Prudhommeaux E., *et al.* (2008). *SPARQL Query Language for RDF*. W3C Recommendation http://www.w3.org/TR/rdf-sparql-query/.

Resnik P. (1995). *Using Information Content to Evaluate Semantic Similarity in a Taxonomy*. IJCAI, Vol 1, pp. 448-453.

Richardson R. and Smeaton A. (1995). *Using WordNet in a Knowledge-based approach to information retrieval*. Proceedings of the BCS-IRSG Colloquium on Information Retrieval.

Rocco D., *et al.* (2005). *Domain-Specific Web Service Discovery with Service Class Descriptions*. ICWS'05, pp.481-488.

Roussopoulos N., *et al.* (1995). *Nearest Neighbor Queries*. In Proceedings of ACM SIGMOD, ACM Press pp. 71-79.

Schenkel R., *et al.* (2005). *Semantic Similarity Search on Semistructured Data with the XXL Search Engine* Information Retrieval (8):521-545.

Shekarpour S., *et al.* (2011). *Keyword-Driven SPARQL Query Generation Leveraging Background Knowledge*. IEEE/WIC/ ACM WI-IAT International Conference, (1)203 – 210.

Shvaiko P. and Euzenat J. (2008). *Ten challenges for ontology matching.* Proceedings of the OTM 2008 Confederated International Conferences pp. 1164–1182.

Spiegler I. (2003). *Technology and knowledge: Bridging a "Generating" Gap*. Information & Management 40(6), 533-539.

Tagarelli A., *et al.* (2010). *Semantic Clustering of XML Documents*. ACM Transactions on Information Systems 28(1):3.

Tagarelli A., *et al.* (2009). *Word Sense Disambiguation for XML Structure Feature Generation*. ESWC, LNCS 5554, pp. 143–157.

Tekli G., *et al.* (2011). *XA2C: a framework for manipulating XML data,* IJWIS Journal, 7(3):240-269.

Tekli J., *et al.* (2011). *A Novel XML Structure Comparison Framework based on Sub-tree Commonalities and Label Semantics*. Elsevier JWS, doi:10.1016/j.websem.2011.10.002.

Tekli J., *et al.* (2012a). *Minimizing User Effort in XML Grammar Matching*. Elsevier Information Sciences Journal, (210)1-40

Tekli J., *et al.* (2012b). *A Novel XML Document Structure Comparison Framework based-on Sub-tree Commonalities and Label Semantics*. Elsevier JWS, (11)14-40.

Terzi E., *et al.* (2003). *Knowledge Representation, Ontologies, and the Semantic Web*. APWeb Conference, pp. 382-387.

Theobald M., *et al.* (2008). *TopX: Efficient and Versatile Top-k Query Processing for Semistructured Data*. VLDB Journal 17:81–115.

Tzitzikas Y., *et al.* (2012). *Blank Node Matching and RDF/S Comparison Functions*. In Proceedings of ISWC'12.

Umer Q., *et al.* (2012). *Semantically Intelligent Semi-Automated Ontology Integration*. World Congress on Engineering, London.

Van Den Heuvel W.J., *et al.* (2003). *Intelligent Web Services Moving Toward a Framework to Compose*. Communications of the ACM 46(10): 103-109.

Van Leuken R. H., *et al.* (2009). *Visual Diversification of Image Search Restuls*. Proceedings of WWW Conference, pp. 341-350.

Vickrey D., *et al.* (2005). *Word Sense Disambiguation for Machine Translation*. EMNLP-CoNLL Conference, 771–778, Vancouver

Yaworsky D. (1992). *Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora*. COLING Conference, Vol 2, pp. 454-460. Nantes.

Ye S., *et al.* (2007). *Automatically Integrating Heterogeneous Ontologies from Structured Web Pages*. International Journal on Semantic Web & Information Systems 3(2):96-111.

Zhang B., *et al.* (2005). *Improving Web Search Results using Affinity Graph*. International ACM SIGIR Conference, pp. 504-511, NY.

Zhang H., *et al.* (2010). *Domain-Specific Web Services for Scientific Application Developpers*. Gateway Computing Environments Workshop (GCE'10) pp. 1-7.

Zhao L., *et al.* (2011). *Mid-Ontology Learning from Linked Data*. COLING'11, pp. 1098-1102, Montreal, Canada.

Zheng Y., *et al.* (2010). *GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory*. IEEE Data Eng. Bull. 33(2): 32-39.

Zins C. (2007). *Conceptual Approaches for Defining Data, Information, and Knowledge*. Journal of the American Society for Information Science and Technology 58(4), 479-493.