Technically Approaching the Semantic Web Bottleneck

Nikolaos Konstantinou*,
Athens Information Technology (AIT)
19,5km Markopoulo Ave.
GR-19002, Peania, Athens, Greece
E-mail: nkons@ait.edu.gr
*Corresponding author

Dimitrios-Emmanuel Spanos,
Periklis Stavrou and
Nikolas Mitrou
National Technical University of Athens (NTUA)
School of Electrical and Computer Engineering
Division of Communications, Electronics and Information Systems
GR-15773, Zografou, Athens, Greece
E-mail: {dspanos, pstavrou}@cn.ntua.gr
E-mail: mitrou@cs.ntua.gr

Abstract: After several years of research, the fundamental Semantic Web technologies have reached a high maturity level. Nevertheless, the average Web user has not yet taken advantage of their full potential. In this paper, we introduce the Semantic Web bottleneck, analyse the main problems that preserve it and suggest ways to overcome it. In particular, we discuss the issues involved in deploying, maintaining and using semantically rich Web applications, decomposing this process into two primal ones: publishing and exploiting semantic content. We analyse the role of key players such as the Web industry, the search engines, the academia, the Web user, and the Web engineers that essentially materialise and use these technologies. A roadmap is provided in order for the Semantic Web to gain further acceptance, based on three major axes: simplicity, mainly entailed by automation, integration with the existing technologies and practices, and adoption by the Web industry driving forces.

Keywords: Semantic Web; Linked Data; Microformats; Annotation; Search Engines.

Reference to this paper should be made as follows: Konstantinou, N., Spanos, D.E., Stavrou, P., and Mitrou, N. (2010) 'Technically Approaching the Semantic Web Bottleneck', Int. J. Web Engineering and Technology, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Nikolaos Konstantinou received his Dipl.-Ing. degree from the School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Greece, in 2004 and his Ph.D. in Electrical and Computer Engineering from NTUA in 2009. Since February 2010, he joined the Autonomic and Grid Computing group in Athens Information Technology (AIT). His research interests include knowledge management in the Web, context-sensitive systems and information flow in the Internet of Things. He is a member of the Technical Chamber of Greece.

Dimitrios-Emmanuel Spanos received his Dipl.-Ing. degree from the Electrical and Computer Engineering School of NTUA and a Master in Business Administration (MBA) from the Athens University of Economics and Business (AUEB), Greece. Since November 2007, he is a PhD candidate in the Multimedia Communications and Web Technologies Research Group in National Technical University of Athens (NTUA), Greece. His research interests include, among others, data integration using Semantic Web technologies, relational-to-RDF mapping and semantic annotation of multimedia content from sensor networks. He is a member of the Technical Chamber of Greece.

Periklis Stavrou graduated in 2008 from National Technical University of Athens (NTUA) with a Dipl.-Ing. degree in Electrical and Computer Engineering. Since October 2008 he is part of the Multimedia Communications and Web Technologies Research Group in NTUA, working towards his PhD. His main interests include Semantic annotation, data fusion using Semantic Web technologies and knowledge representation and reasoning in distributed systems. He is a member of the Technical Chamber of Greece.

Nikolas Mitrou is a Professor in National Technical University of Athens. Prof. Mitrou's research interests are in the areas of digital communication systems & networks (broadband networks in particular) and networked multimedia in all range of studies: design, implementation, modelling, performance evaluation and optimization. Since 1988 Prof. Mitrou has been actively involved in many RACE, ACTS and ESPRIT projects and he was the coordinator of one of them (AC235, WATT). Prof. Mitrou is a member of the IEEE, member of the IFIP WG 6.3 and member of the Technical Chamber of Greece.

1 Introduction: The Semantic Web Roadmap

The main idea behind the Semantic Web remains simple: render Web content meaningful both for human and machine consumption (Berners-Lee *et al.*, 2001). The difficulties arising from the effort of turning this idea into a reality are being extensively investigated for more than a decade. By observing the Semantic Web landscape, we can see that most of the fundamental technologies that constitute its building blocks have reached a mature state. As we will analyse in Section 2, ontology languages, reasoners, ontology editors and other fundamental tools are ready to be launched outside academia and be used in production environments.

Nevertheless, the main idea still remains to a large extent unrealised

(Shadbolt *et al.*, 2006). The so-called Web 2.0 era, consisting mostly of trends such as Facebook, YouTube, Flickr and Delicious to name a few, is dominated by the usage of keywords (or tags). Keywords are popular for annotating published Web content and are also used to perform searches on it. However, the majority of the Web user base has not yet taken notice of the set of semantic technologies emerging in the background leaving semantics out of the casual Web user's everyday experience.

In (Feigenbaum, 1977), the authors, investigating why AI was not gaining popularity outside academia in the late 70's, defined the Knowledge Acquisition bottleneck: “in order for a knowledge based system to be economically profitable, the cost of acquiring and maintaining its knowledge base must be significantly less than the economic benefits derived from its deployment”. At the time, it was simply too expensive (i.e. resource-consuming in terms of time and money) to acquire and encode the large amount of knowledge that an application needed. In analogy, we can observe nowadays and hereby define the contemporary Semantic Web counterpart of the Knowledge Acquisition bottleneck: “it is often too expensive to create and maintain the semantic information an application needs”. Expensiveness is usually raised by lack of automation, error susceptibility or both. By the term semantic annotation we refer to annotations that, in contrast to simple keywords, carry their semantics in the sense that they are commonly and unambiguously understood by software clients since these semantic annotations conform to common vocabularies.

In other words, the process of manually annotating content is still a cumbersome procedure and, admittedly, not an obligatory one, since the benefits of Semantic Web applications still exist more on paper than in real world. This leads though to a vicious circle from which we cannot escape, because such applications need considerable volumes of data to showcase their utility.

The Semantic Web's embryonic nature is reflected in its existing applications. In contrast to the alleged great potential described by the researchers, only a small number of semantically rich on-line applications have made their appearance. An indicative set of case studies by the W3C Semantic Web Education and Outreach (SWEO) Interest Group demonstrates the potential carried by Semantic Web applications in several areas of interest such as automotive, education, eGovernment, energy, eTourism, financial, GIS, and healthcare (`www.w3.org/2001/sw/sweo/public/UseCases/`). Semantic Web technologies are used for various tasks such as content discovery, management and customization, data integration and domain modeling. The common denominator to all these approaches is simplicity. Most of the approaches are “simple”, in the sense that they do not make use of sophisticated semantic features, or, to use a Semantic Web catch phrase, “a little semantics goes a long way”.

Accordingly, as far as it concerns company intranets employing Semantic Web technologies, although they often provide perfectly adequate solutions to a company's needs, they actually fall short of fully exploiting the Semantic Web's exciting potential as a large-scale source of background knowledge (d’Aquin *et al.*, 2008).

Building on the aforementioned observations, in this paper we discuss the reasons why the Semantic Web vision is not yet fully realised and what are the Web's current weak points that need to be strengthened. We analyze the several building blocks that comprise the contemporary Semantic Web in its current state, we analyze their weak points and suggest further directions towards which research efforts should be directed in order to render the technologies beneficial for the key players in the Semantic Web landscape. These key players and the interactions among them are introduced next:

1. *Web users* are the ones who interact with Web pages. Besides surfing, they possibly maintain on-line profile(s) at a social networking site (or more), and post in blogs, fora, and wikis, probably have a homepage and are registered in a number of sites (e.g. IMDB, Wikipedia). Their major gateway to the information that lies in the Web are the search engines. Web users constitute the most passive of all players, nevertheless they can generate valuable content.
2. *Web engineers* are the ones burdened with the task to design Web software systems, implement them and maintain them. In order to complete his/her assignment targeted at fulfilling real-world needs, a Web engineer prefers using mature, robust and reliable techniques and tools. Web engineers are mainly employed by companies and may be at the same time Web developers.
3. *Search engines* are Web applications whose main task is to offer search capabilities. They crawl, index the Web and answer users' queries. Their role is leading because they are the main gateway to information and hence, they are considered one of the more powerful driving forces of the Web. It needs to be emphasised that behind each major search engine implementation there is a privately held company, interested (ultimately) in maximizing its profit. It needs to be clarified that in this paper, we do not include in the search engine definition smaller scale software components that provide local search capabilities, although in some other context, they could qualify as search engines.
4. *Companies* are privately held. In this paper we are mostly focused in companies active in building software or heavily relying on it in order to satisfy their client base and internal needs as well. Such companies are believed to be directly connected to the commercial future of Web

technology.

5 The *Academia* comprises universities and research institutes. The main task of the academia regarding the Semantic Web is basic and applied research into theoretical and technical issues involved in semantic information management and also to drive (together with experts in the Web domain) the development, convergence and adoption of the relevant standards. Most of the technologies serving as Semantic Web's building blocks have evolved out of academia and its role in Web evolution is still active.

Section 2 reviews and elaborates on the fundamental technologies that constitute the Semantic Web. Namely, the knowledge description languages, the rules that can be defined for knowledge processing, the Web Services that allow interoperability among distributed systems, and query mechanisms are investigated. In Section 3 we analyse how semantic content can be published online. This section is structured according to the origin of the content, be it Web documents, relational databases, or other forms of multimedia content. We discuss the difficulties involved in semantically annotating it and suggest ways to overcome them. Section 4 analyses how semantic content can be exploited. The analysis focuses on the difficulties and potential benefits of semantic annotation in search engine implementations and in company environments as well. Section 5 concludes the paper, by summarizing our observations, our recommendations and our main contributing remarks. Focusing on the key players, we describe their actual and desired role in order to bring the Semantic Web to its full potential.

2 Semantic Web Technologies

2.1 Knowledge Description Languages

Mainly, in the Semantic Web ecosystem there are two approaches in describing knowledge: RDFS, a W3C recommendation since 1999 (Brickley and Guha, 1999) and OWL, a W3C recommendation since 2003 (McGuinness and van Harmelen, 2003). Note that due to the openness of the Web, a W3C recommendation is as close to a standard as it can be. This is manifested by the wide adoption of W3C's recommendations by developers and companies alike, although other standard bodies such as OASIS have developed some worth-mentioning standards as well (e.g. BPEL).

The main difference between these two approaches is in the underlying semantics. According to RDF, the perception of the world is modeled as a graph (Brickley and Guha, 1999). An RDF graph is similar to a directed labeled graph, with the difference that RDF allows for more than one (uniquely labeled) edge between the same pair of nodes, the nodes are not necessarily connected to each other and it is allowed to form cycles in the graph. The nodes of an RDF graph contain either resources or literals. The difference between resources and literals is that the latter are not subject to further processing by RDF parsers. OWL brings the expressiveness of logic into the Semantic Web (Horrocks *et al.*, 2003). It adds the formal semantics of very expressive Description Logics, based on standard predicate calculus. Lately, W3C has been working on OWL 2, OWL's successor that since October '09 found its way into a recommendation (`www.w3.org/TR/owl2-overview/`), with new features allowing for more expressiveness and with the definition of different profiles of the language that are effectively subsets of the full OWL 2 Language, which in turn simplify the complexity of the reasoning algorithms (Motik *et al.*, 2009). The role of XML is to provide a syntactic foundation layer for these description languages and the semantic technologies in general.

According to a recent survey regarding the current state of the Semantic Web (Cardoso, 2007), focused mostly in the academia, OWL is used in 75,9 percent of ontology authoring and RDF Schema in 64,9 percent. Therefore, these two approaches in modeling knowledge cover the vast majority of ontology authoring needs. As far as the ontology authoring environment is concerned with, the survey reveals that Protégé (`protege.stanford.edu`), SWOOP (`code.google.com/p/swoop`) and OntoStudio (available online at `www.ontoprise.de`, successor of OntoEdit) cover the majority of these needs, with Protégé being the indisputable dominating solution.

Many large ontologies are currently being developed (or converted from other formats), for instance the National Cancer Institute's ontology, UniProt, BioPAX, ISO 15926, which address specific domains and consist of several thousands of classes. However, not all use complex reasoning; in many cases a small fraction of OWL is used (Dubost and Herman, 2008).

The conclusion that can be drawn by observing the general picture is that the description languages, the ontology authoring environments and applications that employ them (for instance, the W3C SWEO Group case studies mentioned in Section 1) are mature enough. They can be used effectively to solve real-world needs. However, the fact that only a small fraction of OWL constructs is used indicates that OWL's expressivity may be enough for the current application needs. New authoring environments may ease the development of more complex ontologies, but for now, the search for more expressive languages may not be considered as the top priority regarding the Semantic Web state. Instead, simplicity is favoured by application developers, a necessity that led to the development of the RDFS-Plus, a language that, by using a particular subset of OWL, it "… is at the same time useful and can be implemented quickly". (Allemang and Hendler, 2008b). Hence, the role of academia should concentrate in putting effort into pushing and exploring the capabilities of the languages currently offered in collaboration with the industry, rather than researching for more complex terminologies to describe more sophisticated models which would be an overkill for the majority of commercial applications.

2.2 Rules Authoring and Interchange

Rule-based problem solving has been an active topic in AI and expert systems over the last decades. The classic approach to rule-based computational systems comes from work on logical programming and deductive databases. So, how do they fit into the modern Web industry landscape?

Rule engines are not suitable for every application. For many applications, a set of if-then-else statements suffices for the description of the business logic. Rules come into play when this set turns into a complex Boolean logic description and its maintenance and adaptation becomes tedious. In mission-critical production environments, frequently changing parts of the logic can lead to the introduction of errors that could subsequently cause financial losses. In these cases, rule-based approaches can provide concrete, robust solutions.

Conventional rule engine implementations include Jess (`www.jessrules.com`), a non-free rule engine for the Java language and its free counterpart CLIPS (`clipsrules.sourceforge.net`), Drools and JBoss rules (`www.jboss.org/drools`), and the Jena internal rule engine, used to implement Jena's internal reasoners (Carroll *et al.*, 2004). They are all based on the Rete algorithm (Doorenbos, 1995).

From the Semantic Web point of view, we can notice that, since the semantics of OWL follow First Order Logic (FOL), rules can be based on simple Horn rules in the form $P_1 \wedge P_2 ... \rightarrow C$. However, no official W3C recommendation has occurred yet, essentially leaving the choice of a rule language to the application developer. The RIF Working Group has been created for this purpose (Kifer, 2008) and is in the process of developing RIF (Rule Interchange Format) – a core rule language and a set of extensions (dialects) that allow the serialization and interchange of different rule formats. Although RIF can act as a mediator between heterogeneous implementations such as ILOG JRules (`www.ilog.com/products/jrules/`), Oracle Business Rules and Prova (`www.prova.ws`) (Hallmark *et al.*, 2008), it has not matured enough (W3C candidate recommendation from 1st October 2009) and may need additional rules-with-actions dialects and datatypes and built-ins definitions.

Currently, inside the Semantic Web community, RuleML (`www.ruleml.org`) and SWRL (Horrocks *et al.*, 2004) are the most popular representatives of the kind. RuleML is a Semantic Web rule description language, offering flavors of the same language such as the XML/RDF combining, the RDF-only, and the FOL RuleML, while SWRL is a W3C member submission that combines OWL with RuleML. SWRL combines OWL DL and OWL Lite with Unary/Binary Datalog RuleML, in order to allow the combination of Horn-like rules with OWL knowledge bases. Supporting only unary and binary predicates, without disjunction and functions, SWRL is less expressive than its RIF counterpart, the basic logic dialect RIF-BLD.

Another approach for the definition of a rule language is through the use of SPARQL `CONSTRUCT` clauses. An extension to this approach is the SPARQL Inference Notation (SPIN) (`www.spinrdf.org`), an RDF Schema for SPARQL, that allows domain modelers to attach inference rules (as SPARQL `CONSTRUCT` queries) and constraint checks (as `CONSTRUCT` or `ASK` queries) to RDFS or OWL class definitions. Although this seems like a simple solution, construct queries using the full features of the non-monotonic SPARQL language (especially the combination of left join and union, which is the main source of complexity (Perez *et al.*, 2006)), result in an expressive and complex rule language (non-recursive Datalog with negation (Schenk, 2007)), and its combination with ontologies needs to be further studied (Polleres, 2007).

The difficulty behind recommending a rule solution for the Semantic Web lies mostly in balancing the required expressiveness with proper algorithm computational properties, such as termination in polynomial time or decidability (the second property being broader than the first one). Still, reasoning systems such as Pellet (Sirin *et al.*, 2007) support SWRL but not in its full extent. Therefore, the first priority in the related research efforts should be a convergence that will lead to a W3C recommendation, in order for the rules to “graduate” the academic environment.

Keeping in mind the directions towards which actions should be taken in order to assist the Web engineer, we can note that it is important that more technical spaces be incorporated in rule definitions, in the sense of the XML/RDF combining RuleML. Instead of taking into account only semantic resources, rule engine implementations should embrace relevant technical fields such as relational databases, Web Service messages, or even arbitrary programming language code snippets. This expansion can turn them into powerful tools in the hands of a (Web) developer. The Web developer is not willing to sacrifice in the altar of expressiveness the power offered by traditional rule engines in declarations such as in the following pieces of pseudocode, presented in the event-condition-action (Papamarkos *et al.*, 2003) pattern:

```
on [incoming Web Service request]
if [some property value in the ontology model]
then [modify XML file]
```

or

```
on [tuple insertion in a database table]
if [code snippet]
then [insert a triple in the ontology model]
```

or

```
on [triple insertion in the ontology model]
if [boolean condition checking the inserted triple]
then [code snippet]
```

The examples above demonstrate the conception of integration with existing technologies that should be available to the Web developer. Such implementations, embracing different technical domains, are necessary in order for the rules to provide functionality capable of tackling more realistic scenarios. One, for example, could easily portray technical-space-spanning rule-based applications that semantically annotate data generated from sensor networks (Konstantinou *et al.*, 2010). In the context of such applications, it is critical to be able to perform event-based annotation, in other words carry out the annotation task as soon as there is a new measurement taken from a sensor. Furthermore, it would be highly convenient if the antecedents and consequents of an application's rules could be referring to different technical spaces, for instance the body of the rule could contain a condition checking an ontology model while the head could comprise a XML-file modification or an insertion statement in a database. Moreover, this kind of rules does not invalidate previous familiarity with relevant technologies but these technologies are rather utilised in order to design and implement more intelligent and robust solutions. Without this integration, semantic rules are deemed to isolated, case-specific solutions with restricted potential.

2.3 Semantic Web Services

Over the last years, the prevalence of the Web Service technologies in combination with the increasing needs of the industry for distributed, interoperable and integrated systems has led to the adoption of Web Services as the common currency for service-oriented computing. In their current state, Web Services present competent technical properties such as modularity, scalability and reusability. They can lead to the realization of protocol-independent, coarse-grained, and loosely coupled architectures. For the enterprises, the main advantage is that they can hide implementation details from external modules, allowing them to build their services without interacting with other developers or clients prior to execution time.

Building upon Web Services' merits, Semantic Web can add intelligence in service-oriented architectures. Semantic Web Services are built on top of the conventional Web Services by extending their description of functional behaviour. A semantic description of a Web Service's functionality enables intelligence in orchestration of composite workflows and negotiations, needed in industrial environments. The main idea is to preserve current technical and functional characteristics and current successful standards (SOAP, WSDL, UDDI) and target exclusively the description.

The first attempt of a related standard is OWL-S (a successor to DAML-S in the same way that OWL is a successor of DAML), a W3C member submission since 2004 (`www.w3.org/Submission/OWL-S/`), but not a recommendation yet. It is designed to assist automated service discovery, composition and invocation, by targeting the service description. Another candidate is SWSF, a W3C member submission since 2005 (`www.w3.org/Submission/SWSF/`). SWSF comprises two components, the Semantic Web Service Language (SWSL) used to specify formal characterizations of Web Service concepts and the Semantic Web Service Ontology (SWSO) that presents a conceptual model by which Web Services can be described. Finally, WSMO (`www.w3.org/Submission/WSMO/`) provides an ontology for describing the core elements of Semantic Web Services. The main concepts of the WSMO approach are ontologies describing the relevant aspects of the domains of discourse, web services providing formal descriptions of the interfaces and capabilities of a web service, goals that present user desires, and mediators, which represent elements that overcome interoperability problems between different WSMO elements.

Technically, the Semantic Web effort targets the Web Service description, the WSDL document (hence WSDL-grounding) or the corresponding WADL, a proposed description for Web Services that follow the RESTful architectural style. The goal is to annotate the functionality offered in order to leverage Web Service choreography and business process execution to an upper level. In (Belhajjame *et al.*, 2008), an approach is presented for semi-automatically annotating Web Services, showing that the use of annotations considerably increases the number of services located by discovery queries even with a small starting set of annotations.

In practice however, this is a highly error-prone task. Combined with the fact that in industrial environments, priority is given to robust, optimised code tailored to suit specific and possibly complex and rapidly evolving needs, annotation comes second. What we suggest is that simplicity is added in the annotation process by leaving the task up to the main responsible: the developer. Annotations should be included in the source code (e.g. C, Java) itself and not be provided by external tools and configuration files. But, in order to advance towards this direction, a standard is needed in the first place. The adoption of a common standard will open the path for tool vendors to offer to the developer what is lacking from current implementations: automation, that will entail simplicity.

2.4 Semantic Queries

Like search engines have become over the years the driving force that rendered the World Wide Web an invaluable tool, semantic queries are believed (or hoped) to play the same role for the case of the Semantic Web. The standardization of SPARQL (Prud'hommeaux and Seaborne, 2007) is a key step towards this goal, since most application developers and vendors are compelled to support it, instead of using proprietary languages and formats, as was the norm not so many years ago. SPARQL is a W3C recommendation since 2007 and denotes a family of standards including a query language for RDF, a protocol definition for sending SPARQL queries from a client to a query processor and an XML-based serialization format for results returned by a SPARQL query. However, the term SPARQL is almost exclusively used to refer to the query language.

SPARQL, unlike earlier query languages proposed that traverse the RDF graph (e.g. RQL (Karvounarakis *et al.*, 2002), available in the Sesame system (Broekstra *et al.*, 2003)), does not take into account the graph level, but instead models the graph as a set of triples. Essentially, in a SPARQL query, a graph pattern (i.e. one or more triple patterns) is specified and nodes that match this pattern are returned (i.e. URIs or literals). The SPARQL syntax bears a lot of similarities to SQL, the `SELECT FROM WHERE` syntax being the most striking one. However, it is still a long road until SPARQL reaches the maturity level of SQL and satisfies most of today users' needs.

In its current form, SPARQL is merely a way to access raw data (URIs in this context) from an RDF or OWL graph, letting the user do the result processing. However, as RDF and OWL promise to model every information and data available on the Web, leading to an eventual integration of all imaginable sources, SPARQL will be the gateway for querying information and knowledge. Thus, it is rational to expect from SPARQL at least as many features as SQL currently supports, if not more. Unfortunately, this is not the case for the current SPARQL recommendation, as several omissions have been reported (Konstantinou *et al.*, 2008; Weiske and Auer, 2007).

Grouping and aggregation, functionalities present in SQL in the form of the GROUP BY operator and MIN, MAX, SUM, COUNT or AVG functions respectively, are not supported in SPARQL, while the sort operator ORDER BY can be applied only on a global level and not solely on the OPTIONAL part of the query. Support, even for simple mathematical calculations, does not extend beyond basic operations; the inclusion of trigonometric functions or exponents could prove useful, especially in the context of geographic information systems.

Furthermore, in contrast to SQL, SPARQL does not support nested queries, hence you cannot search a result set (in other words, SPARQL does not allow a CONSTRUCT query in the FROM part of a query, where the graphs to be queried are specified). Another missing feature of SPARQL is the functionality offered by SELECT WHERE LIKE statement in SQL, allowing for keyword-based queries. Of course, SPARQL offers the regex() function for string pattern matching, but it cannot emulate the functionality of the LIKE operator. As mentioned earlier, SPARQL only allows for unbound variables in its SELECT part, therefore rejecting the use of functions or other operators; this restriction renders SPARQL an elementary query language where only URIs or literals can be returned, while in practical use cases, users opt for (some) result processing. The list of possible additional features in SPARQL could include stored procedures, updates, insertions, and deletions of the underlying graph as well as triggers on these actions. Already popular among Semantic Web developers is SPARQL/Update (SPARUL, jena.hpl.hp.com/~afs/SPARQL-Update.html), an extension of SPARQL included in Jena (Carroll *et al.*, 2004), the leading Semantic Web development framework, allowing for the update, creation or removal of RDF graphs.

Regarding security, we could state that this is a somehow neglected aspect of the Semantic Web; thus, it comes as no surprise that SPARQL does not take care of security issues at all. The inclusion of prepared statements that speed up execution of similar queries and deal with the problem of SQL injection attacks could benefit SPARQL in the same way as they do in SQL. Additionally, transactions (START TRANSACTION, COMMIT, ROLLBACK) that are widely used in commercial applications and in applications where security is a crucial factor are still absent from the SPARQL formal specification. Finally, named graph restriction (for all or selected users and query clauses pairs) is a feature that would solve simple security issues, like keeping private data of the store (e.g. email addresses, credit card numbers) safe.

Nevertheless, SPARQL embodies a variety of interesting features not present in SQL. A feature that can be met in almost all of the query languages for RDF is the use of the OPTIONAL operator that does not modify the results in case of nonexistence. This is equivalent to a left outer join in SQL, but the SPARQL syntax is much more intuitive and user-friendly in this case.

SPARQL should be enhanced with at least some of the above features in order to become the prevalent query language in the next generation Web and the SPARQL Working Group is already working on incorporating some of them in the next SPARQL version (`www.w3.org/TR/sparql-features/`). Examples of the latter case include the query engine of OpenLink's Virtuoso (`www.openlinksw.com/virtuoso/`), which extends SPARQL with aggregates, nesting and subqueries, allowing the user to insert SPARQL queries inside SQL, the slightly enhanced SQL version of Oracle 11g (Lopez and Das, 2007), SPASQL (`w3.org/2005/05/22-SPARQL-MySQL/XTech`) which offers a similar functionality, embedding SPARQL into SQL, LARQ (`jena.sourceforge.net/ARQ/lucene-arq.html`) that integrates SPARQL with Apache's full-text search framework Lucene and the SPARQL+ extension of the ARC RDF store (`arc.semsol.org/docs/v2/sparql+`) which offers most of the common aggregates and extends SPARUL's `INSERT` with `CONSTRUCT` clause. In conclusion, regarding the future of SPARQL and in order to become a more powerful tool in the hands of a Web engineer is its integration with related technical spaces in a way similar for instance to SQL implementations that support nested XML queries.

3 Publishing Semantic Content

Since the definition and introduction of the term Semantic Web in the limelight, much effort has been placed on the development of new standards, technologies and applications which would hopefully convince the large public of the Semantic Web's utility and superiority compared to the current and “obsolete” World Wide Web, as supported by the former's afficionados. However, for the Semantic Web to become a reality, the ones that need to adopt the new technologies are common everyday Web users, who tend to prefer simple solutions; this was one of the reasons of the success of the World Wide Web, to start with. Moreover, the expansion of the Semantic Web will be realised when the generation of Semantic Web content becomes a trivial matter for the average Web user in the same way that the effortless design of a Web document has led to the rapid growth of the Web. In the case of the World Wide Web, documents constitute the fundamental building block, while for the Semantic Web, data – however generic and vague this might sound – plays this role. Hence, we argue that there has to be a shift in the attention of the Semantic Web community towards ways to render the generation of Semantic Web content more convenient and less cumbersome than it is today.

Throughout this paper, we refer to the process of creating, managing and administrating content using the term “publishing”. In this Section, we classify this generation of semantic content according to the data source: whether it is Web documents, relational databases, or multimedia. We present the theoretical and technical difficulties of turning data, in each of the above cases, into “Linked Data”, a term already popular among Semantic Web followers that is briefly described next.

3.1 Linked Data: Implementing the Semantic Web Vision

The emerging “Web of Data” concept is to materialise the Semantic Web vision, to advance from a Web of Documents to a Web of (Linked) Data (Bizer *et al.*, 2008). Instead of the current experience where the users navigate among (HTML) pages, the main idea is to navigate among (RDF) data. In other words, just as the current Web can be crawled by users (and search engines) through hyperlinks, the Web of Data can be crawled through RDF links. RDF links simply assert relationships between Web resources by forming triples according to the Semantic Web paradigm: (*resource*, *property*, *resource*) and the main difference from simple hyperlinks is that, unlike the latter, they possess some meaning.

There are several ways through which Linked Data can be materialised on the Web. RDF data can originate “on the fly” from Web pages through microformats and GRDDL, RDF middleware, RESTful Web Services, or community-driven Linked Data projects (`www.linkeddata.org`).

The technical approach is quite straightforward. Resources are uniquely identified using URIs and links between them are RDF links. Common vocabularies can be used to represent information (DC, FOAF, SIOC, DOAP, SKOS, CC etc.), and there is no longer need to define new vocabularies from scratch. Therefore, common HTTP servers can be used to serve RDF graphs, simply by adding the respective MIME type declaration, for instance `"application/rdf+xml"` or `"text/rdf+n3"`, in a Web page's `<head>` section.

The convenience behind this trick is that unknown tags are simply ignored by the browser while rendering an HTML page. Therefore, a Web page can keep its current form, while its semantic annotations can be exploited by a Linked Data browser or a RESTful Web Service. As far as it concerns the Web developer, the majority of the modern development frameworks that follow the MVC (Model- View-Controller) pattern (for instance JSF, .NET, Struts or Tapestry) allow insertion of such declarations in the View part of the application and thus, easily support semantic description without need of modifying the supporting code. From the Web user's point of view, Linked Data browsing and regular Web page browsing should be seamless, so that the burden of installing additional browsers or plugins is kept at a minimum level, letting benefits outweigh the (time) costs.

Although, as stated in the previous section, most of the technologies were stable since 2006, the main drawback for accomplishing Semantic Web was the fact that most of the use cases were restricted to closed-world environments. Motivated by the absence of real world web-scale scenarios, the Semantic Web community, and particularly, the Linking Open Data community project, started the effort of publishing RDF-based data. Today, having an increasing number of public datasets, innovative Linked Data applications should be the next step.

Linked Data-driven applications are grouped into four categories (Hausenblas, 2009): (a) *Content reuse applications*, such as BBC's Music store that (re)uses metadata from DBpedia, and MusicBrainz (`www.musicbrainz.org`), (b) *semantic tagging and rating* applications such as Faviki (`www.faviki.com`) that uses unambiguous identifiers from DBpedia, (c) *integrated question-answering systems*, such as DBpedia mobile (Becker and Bizer, 2008) able to indicate locations from the DBpedia dataset in the user's vicinity, and (d) *event data management systems*, such as Virtuoso's ODS-Calendar (`virtuoso.openlinksw.com`) that can organise events, tasks, and notes. Using the Linked Data approach, Data Webs are also expected to evolve in numerous fields from biology (Zhao *et al.*, 2009) to software engineering (Iqbal *et al.*, 2009).

In order to ease the design of Linked Data applications, a standard procedure must be adopted in publishing and consuming datasets. VoiD (Alexander *et al.*, 2009) is a vocabulary for describing a dataset's content, accessibility, licensing and the links it holds to other datasets. Enriching datasets with VoiD terms, will simplify the design of data mashup applications. For consuming unstructured or semi-structured data from popular Web 2.0 sites, various RDF wrappers exist, such as for Flickr (`apassant.net/home/2007/12/flickrdf/`), and Delicious (`linkeddata.uriburner.com`). This great amount of data, together with the community maintained datasets, constitutes the current Web of Data (the so called LOD cloud) offering a huge data/resource layer for many applications. SPARQL, in a sense, plays the role of a RESTful API for the Web of Data. However, there is a main disadvantage in its current state: it is a read-only API, meaning that manipulation of datasets, directly through the RDF-based environment (the Linked Data application), is not provided. Modifications are carried out, either directly by the dataset administrators or in case of Web 2.0 sites/services through the use of their specific API. SPARQL/Update is a direct solution to the former case and can be mapped, as described in (Ureche *et al.*, 2009), to site specific APIs function calls, for the latter. SPARQL and its extension SPARQL/Update, are not purely RESTful services. As explained in (Wilde and Hausenblas, 2009), with the use of named graphs, the RESTful mapping of the SPARQL protocol, depends on the decision of identified resources, which could possibly be the information units (e.g. books, documents), all the subjects of the RDF graph, or every triple. It is clear that there is still work to be done for establishing standard procedures and moving towards wider adoption of the Linked Data principles. For designing more critical applications also, issues like trust and data provenance, guided or automated link creation and maintenance and data fusion from sensors and other content-generating devices, have to be studied.

3.2 How to Transform Web Documents to (Linked) Web Data

The World Wide Web in its present form is a Web of Documents. This makes Web documents a principal source of data and information. They are unstructured data, since they offer a textual or sometimes graphic representation of information, understandable by the human user, but (almost) useless when it comes to being processed by a software agent.

In order to produce dynamic content, sophisticated Web applications usually follow the so-called n-tier (usually 3-tier) architectural approach. More abstractly, 3-tier applications can be logically viewed as consisting of the presentation, the (business) logic and the data tier (Edwards and DeVoe, 1997), with tiers often referred to as layers. In this context, Web documents belong to the presentation layer, are retrieved in HTTP protocol by the user's browser and they represent more or less the “tip of the iceberg”. Hence, the process of enriching a Web document by embedding semantic information in it can be considered as enhancing the presentation layer; for example, instead of writing:

```
<div about="http://www.example.org/jdoe">
...
<p> My nickname is John D. </p>
</div>
```

we can write

```
<div xmlns:foaf="http://xmlns.com/foaf/0.1/"
about="http://www.example.org/jdoe">
...
<p> My nickname is <span property="foaf:nick"> John
D. </span>. </p>
</div>
```

The output in the Web browser is the same as before; the only difference is that, with the second choice, we have generated the following RDF triple (in Turtle notation) where the subject of the triple is the URI denoted by the about attribute:

```
<http://www.example.org/jdoe> foaf:nick "John D.".
```

The above example uses RDFa, a W3C recommendation that uses existing XHTML attributes and introduces some new ones as well that define semantically elements or parts of the original Web document. RDFa can be easily mistaken for a microformat (Khare, 2006) by a less experienced user, while, in fact, these two technologies present significant differences. Microformats, such as hCalendar, hCard and hCal have been introduced prior to RDFa to serve the same purpose, that of semantic markup. Microformats, in a way similar to RDFa, use existing XHTML attributes in order to provide data with semantics; however, microformats do not use URIs, since every microformat uses its own predefined vocabulary. Hence, microformats do not provide a unambiguous semantic representation of data, let alone the fact that they cannot be easily combined in a single Web document, given that every single microformat uses different XHTML attributes in, possibly, conflicting ways with each other.

The main advantage, though, of microformats, is their expansion, as they have been embraced by the Web community and are already part of millions of Web pages. This large volume of markup, should not be ignored and therefore, efficient scalable methods that would transform it to an equivalent semantic representation are being sought, with the most prominent ones being the use of GRDDL, a W3C recommendation since 2007 (Connolly, 2007), to transform XHTML pages containing microformats into RDF or the use of hGRDDL (Adida, 2008) transforming microformats directly to RDFa in a way similar to how XSLT can transform XML documents. These microformats can reside in XHTML documents or other XML formats such as Atom or RSS. The main advantages of RDFa over microformats' simplicity are its extensible vocabulary and independent syntax. Eventually, the choice between microformats or RDFa is left to the content provider and it depends on whether his needs are covered by the domain-specific vocabularies of microformats. In fact, this rivalry is a typical "de jure" vs. "de facto" standard competition, with microformats being the choice of Web publishers so far and RDFa being promoted by the W3C.

From a Web user's perspective, microformats and semantic markup can help in creating, personalizing, sharing and reusing content. In (Ankolekar *et al.*, 2008), a visionary example is provided about how a casual Web user could use these technologies in order to improve her online presence, with the authors concluding that "the need of the hour is to focus on more simple

Web application scenarios”.

From the Web developer's point of view, microformats can easily and conveniently be part of the View element (.aspx and .ascx for .NET, .jspx for JSF, .tml for Tapestry and so on) of modern MVC pattern-oriented Web development frameworks. In other words, they can and should be part of Web applications, even when the use of Semantic Web technologies is tentative. Web applications that use microformats can range from simple Web sites to large-scale complex business systems; there is virtually no restriction.

3.3 Targeting the Database

Static Web documents are not the only source of data in the Web and, contrary to popular belief, they do not constitute the main data source in the Web. The majority of data in the Web resides in databases and are only visible as content of a dynamic Web document, generated in reply to a user's request. Dynamic Web documents constitute a part of the Web, known as the Deep Web (Bergman, 2001), where methods of annotation portrayed in the previous section do not apply. To expose this data in the Semantic Web and make them part of the Linked Data ecosystem, there are in fact two ways: either use a middleware product or service to export the entire or a portion of the source database schema or annotate directly a dynamic Web document that contains some database-extracted values.

Starting with the latter and referring to the example of the previous section, we could say that the methods of annotation described are in fact manual, in the sense that they cannot be reproduced to other HTML files or be automatically linked to other Web data without human intervention. Manual annotation is expensive in all aspects and possesses very limited potential. Let us see what happens when more dynamic results are needed, in the usual case where the content of an HTML tag is populated dynamically from database records, and automation is needed. For instance, a Web application in JSF that retrieves the names of some cities from a database backend would look like:

```
<h:outputText value="#{client.address.city.name}"/>
```

In order to automatically annotate the city name in this example, we would imagine an approach such as the following, where the `client.address.city.SKOSreference` is an example of a Java class attribute maintaining a liaison to a URI provided by SKOS that corresponds to the city of the example:

```
<h:outputText value="#{client.address.city.name}"
```

```
    rel="#{client.address.city.SKOSreference}"/>
```

Such an approach would provide the necessary simplicity, readability and, most importantly, automation to the developer. Without such an approach, it is practically infeasible to annotate (correctly!) a set of e.g. several thousands of cities. However, these approaches are still missing from the state-of-the-art tools and frameworks available to a Web developer; he/she will have to modify accordingly the Business Logic Layer.

We argue that such functionality for automated content annotation should be part of a tool, a framework or a code library. A relatively easy way to achieve automation would be with the use of code annotations (as in Java annotations). Direct mapping of Java classes to RDF concepts is a novel technique that simplifies the annotation procedure. Annotations are widely used, for instance in Hibernate (`www.hibernate.org`), a highly popular Object-Relational mapping framework for J2EE environments. In a similar fashion, further acceptance of object-to-class mappings such as Jersey (`jersey.dev.java.net`), would be easily adopted by developers, regarding existing knowledge in software houses.

Moreover, extensions of the current Web development frameworks would be beneficial to the developer. For instance the JavaServer Faces technology, part of the widely respected J2EE industry standard, uses its namespace to create HTML components. What would be needed, for semantic extensions to be provided to the developer, are extra tags and code libraries that automate the content annotation task as in the example above.

The other choice of generating semantic content from a database is, as stated at the beginning of this section, using some middleware solution. “Middleware” is a widely used generic term that in general refers to a system purposed to be the intermediate between two software systems (or conceptual layers, when referring to a generic architecture). In the Semantic Web context, the term refers to a software component intermediating between data on one side and semantic knowledge, stored as an ontology model or triples on the other side.

Numerous commercial middleware solutions have been proposed, to address mainly the issue of integration of heterogeneous data sources and interoperability among different data formats. The common denominator of these solutions are SPARQL endpoints, which serve as a gateway to the underlying data and the knowledge emerging from its mapping to a unifying model. The majority of these tools extend their capabilities and features even beyond the functionality of a traditional middleware, since they offer storage for both structured and unstructured data, they typically include one triple store for storing all the metadata information, they are able to perform tasks such as versioning and access control, and they advertise themselves as integrated solutions performing data management and integration. Some, such as OpenLink's Virtuoso (`www.openlinksw.com/virtuoso/`), even perform process integration and Web services composition. Furthermore, these tools come in many flavours: commercial (Virtuoso (also offered in a limited functionality open-source version), Profium Metadata Server (www.profium.com)), open-source (Open Anzo (`www.openanzo.org`)) or in the form of SaaS (Software as a Service) in the case of the Talis Platform (`www.talis.com/platform/`). The question remains though: why do these solutions and services are only used by a tiny portion of the Web community, retaining the Web in its "non-semantic" form?

An important impediment is the, often non negligible, difficulty in learning and using these tools, especially for the case of the Web developer, who is not willing or does not have enough time to engage himself/herself in RDF or SPARQL tutorials. Hence, such tools need to place simplicity and user-interface friendliness as their future priorities, if they want to gain a larger user base.

Performance is another key factor; answering SPARQL queries and inferencing over ontologies with millions of individuals (or more) is a task of high complexity, requiring considerable processing power. Fortunately, the advent of cloud computing and the increasing availability of services offered "in the cloud", such as Amazon's Elastic Compute Cloud (EC2) or Google's App Engine, and frameworks for cluster programming such as the open source Apache Hadoop project (hadoop.apache.org) alleviate this problem, although it is not clear how divide-and-conquer approaches can be successfully used in Semantic Web problems such as reasoning (Oren *et al.*, 2009).

Nevertheless, the use of an integration platform such as the ones mentioned above is not the only way we can make data available for use in the Semantic Web context. As pointed earlier and as estimated by a relatively recent study (He *et al.*, 2007) citing that the Deep Web has increased in size approximately by a factor of 7 since 2001, the main bulk of Web content is located in databases and to a large percentage, is not covered by current search engines (according to (He *et al.*, 2007), two thirds of this data remains uncharted territory). Thus, it comes naturally that a lot of research effort has been placed in trying to link relational databases to ontologies, for an, as automated as possible, translation of data in semantic content. An approach or tool capturing as much of the semantics of a database with respect to a particular domain ontology and proposing possible correspondences between the two models would be the key to unleashing large volumes of data in the Semantic Web, given that the owner of each database grants the appropriate permissions. In a nutshell, such a tool would consist mainly of a SPARQL-to-SQL query translator, allowing for the conversion of incoming SPARQL queries to SQL queries that can access and retrieve the data. Among the most notable efforts in the area are D2RQ (Bizer and Seaborne, 2004), SquirrelRDF (`jena.sourceforge.net/SquirrelRDF/`), Virtuoso's RDF Views (Blakeley, 2007) and Triplify (Auer *et al.*, 2009).

Again, the main drawback of these solutions is the fact that they are destined to be used by a developer or a user with some basic familiarity with programming and Semantic Web notions. One could argue that the largest amount of data resides in corporate or governmental databases, hence it is rational that the database administrator, who is aware of the database schema semantics, will use these components to derive Linked Data. However, the aforementioned administrator or developer will rarely fully possess the underlying knowledge about the specific application domain the data refers to. Hence, the process of linking a database to an ontology must involve a domain specialist as well, who will specify the correspondences and does not need to have any programming skills; in such cases, the presence of a user-friendly graphical or programming interface in a mapping tool can be considered indispensable.

3.4 Multimedia Content

Multimedia content typically refers to audiovisual streams, pictures, 3D objects and geographical information. Even simple text can be considered as multimedia content, since it is encoded into bit streams. The important observation is that multimedia content is usually stored in a way that needs to be reproduced by appropriate software in order to be human-understandable. Therefore, metadata annotation is important in order to render it useful for human consumption. In other words, it is impossible to search in a multimedia repository for specific information that might be present but can be lost without appropriate annotation. Multimedia repositories greatly need correct annotation, without which their usability falls dramatically.

This necessity, in combination with the Semantic Web prevalence led to the creation of a series of tools that allow the semantic annotation of multimedia files, manually in most of the cases, usually aided by semi-automatic metadata extraction techniques. These tools include for instance Vannotea (Schroeter *et al.*, 2006) that can annotate collections of multimedia files, and M-Ontomat Annotizer (Petridis *et al.*, 2006) that can link low-level MPEG-7 visual descriptors with RDF(S) ontologies.

Also, Semantic Web-compliant standards have been proposed such as the OWL-based VERL and VEML (Francois *et al.*, 2005). These standards are employed in order to annotate and record objects and (sub)events in video streams. Another approach is Adobe's open, standards-based XMP (`www.adobe.com/products/xmp/`).

Regarding online multimedia annotation however, tagging is preferred.

Social tagging sites, such as Delicious, Flickr and YouTube, constitute a popular and easy way to annotate multimedia, such as images and videos. Another buzzword used alternatively for collaborative tagging is folksonomy (a term combining the words folk and taxonomy). On a more formal basis, folksonomies are a set of triples of the form {actor, tag, object}, representing the fact that a user has annotated some object with a tag. Therefore, folksonomies include a social dimension as well (Mika, 2007).

Tagging pictures in one of the above sites does not entail the generation of semantic content, since users may use an open vocabulary to annotate images and tag search is based on standard keyword search. That is one of the reasons why we currently can have collected, but not collective knowledge (Gruber, 2008), i.e. emergent knowledge deriving from inference over known facts, instead of simple aggregation of knowledge, in the form of e.g. a tag cloud. Even if restrictions to the tags that may be used are applied and all tags are ensured to be described by a URI, this annotation still fails to be categorised as "semantic", since the predicate linking the subject (e.g. the URL of the image being tagged) to the object (e.g. the tag URI) is missing. Flickr has already initiated machine-tags, in which the predicate belongs to a well-known massively used vocabulary, such as FOAF or SKOS; they could serve as an example to other applications or services seeking to facilitate the generation of semantic content. There are already some running projects and initiatives, studying the issue of bringing semantics to the tags, such as the MOAT: Meaning Of A Tag Project (Passant and Laublet, 2008) and TagCommons (`www.tagcommons.org`).

Once again, we can observe that the state of multimedia annotation is the same with the Web of documents: while tools exist and relevant standards have been developed, the majority of online practices prefer simple "tagging" to semantically annotating content. It appears that the Semantic Web poses a big knowledge overhead to the Web user. Therefore, the need in this direction is the production of interfaces that can automate the annotation procedure without requiring expertise in the related technologies.

4 Exploiting Semantic Content

In the scope of this paper, the term "exploiting" refers to the action of effectively querying and retrieving semantic content, or integrating it with other sources or further processing it in order to extract conclusions and leverage its conducted value in general. Thus, in this Section we analyse the actual and the potential way in which existing information can be exploited in a way beneficial for the end user, the Web engineer and the industry, while we justify and analyse the leading role of search engines.

4.1 The Role of Search Engines

Search engines constitute a gateway to the information provided in the Web. Everyday experience indicates that, without them, the Web content would be so difficult to be found that to a large extent it would be useless. Exabytes of data in millions of Web pages are impossible to be searched without the use of special Information Retrieval techniques offered by the current search engine implementations. This reality renders search engines a driving force for the evolution of the Semantic Web and the Web itself.

The fact that users typically prefer to modify their query instead of navigating in the search engine result pages is an indication of the importance and gravity of first page results. This, in turn, impacts the way in which most industry Web sites publish their content. “Search engine friendly” is a term widely known and respected in the industry. Based mainly on keywords and content, search engine optimization means increasing a Web site's visibility in the search engines (mainly Google, Yahoo! and Microsoft). In (Zhang and Dimitroff, 2005a,b), a comprehensive survey is presented concerning the factors that impact this visibility. These factors include, for instance, the metadata structure, the content, and the hyperlink cited status. However, since search engines are private companies, they are not usually willing to disclose in-house intellectual property, and the conclusions are mostly based on observations.

Additionally, semantic search engines targeting only at semantic content have been proposed lately, such as Swoogle (`swoogle.umbc.edu`) that stores semantic documents in a way similar to Google, Falcon-S (`www.falcons.com.cn`), Sindice (`www.sindice.com`), SWSE (`swse.org`), OntoSelect (`olp.dfki.de/ontoselect`) and Watson (`watson.kmi.open.ac.uk`) that offer a functionality similar to Swoogle, or ORAKEL (Cimiano *et al.*, 2008) that (as Watson) processes natural language. There are already various prototypes around. Nevertheless, none of them seems to threaten the dominance of the conventional search engines over the users' preference as a gateway to Web information. In addition, it can be safely observed that semantic extensions are still missing from the mainstream, publicly available search engine implementations.

The dominant search engines, for a long period, have been reluctant to adopt semantic technologies. For example, in (Halevy *et al.*, 2009), Google engineers argue that there are two approaches in analysing content – the “semantic” and the “statistical” analysis – and they seem to be in favor of the statistical analysis for a series of reasons. Yahoo! on the other hand, was the first to offer the SearchMonkey and Microsearch (Baeza-Yates *et al.*, 2008; Mika, 2008) extensions of their search engine that recognise many Semantic Web vocabularies and support RDFa and various microformats (hCard, hCalendar, hReview, hAtom, hResume, adr, geo, tag, xfn, etc.).

If the search engine giants behave in favour of semantic knowledge, a great boost in the semantic technologies will be given. As claimed in (Hendler, 2008), among the most important problems that hold back the Semantic Web evolution is that companies “are reluctant to implement products until they see a market forming, but the market does not tend to form until the tools are available”. In this sense, companies venturing on-line are not willing to publish semantic content, since it does not entail any direct benefit regarding their Web site's front, and search engines do not wish to invest in yet immature advanced technologies. Therefore, we can safely deduce that the commercial future of the Semantic Web is bound to search engine optimization.

Furthermore, exploiting semantic information residing in databases – and not only in Web documents – can offer a solution to the problem of searching the large volumes of data on the Web that are stored using relational database technology. A manageable way of bringing to light this data is circumventing the Web application and attack directly the problem's source: the database. Middleware and triplestore solutions offer an efficient solution. Parallel to a Web application's functionality, views over its data can be exposed through a respective SPARQL endpoint. We should note here that the owner of the database will be the one who defines these views, in other words, the one who decides which part of its data goes public and which not. Multiple benefits arise from such an approach, for every player included:

1 Search engines can multiply the indexed content and serve more efficiently their purpose: deliver more accurate (more sound and more complete) results to the end user/searcher. As soon as a search engine incorporates semantic technology to better serve its customer's (i.e. the end user) needs through more personalised and accurate results, it is highly probable that it will gain a considerable share in the search engine market. Eventually, the rest of its competitors will follow in order to eliminate the competitive advantage of the innovator, leading us into a “no turning back to keyword search” road. Also, if search engines take into account semantic metadata, companies will have a stronger motivation to publish semantic information and the commercial future of the Semantic Web will look brighter. This strengthens our belief that the Semantic Web's commercial future is connected to its adoption by conventional search engines. If the semantic annotation becomes a synonym with search engine optimization, a great boost will be given to the Semantic Web, in addition to the benefits for the end user and the Web itself.

It should be kept in mind that the steps towards more semantic search engines should not abolish the convenience already offered in terms of simplicity and performance. Semantic search engines should not impose an additional knowledge overhead to the user. They should rather simplify his/her effort for finding accurate information. Additionally, in Web searches, the users are not willing to wait; search times should be kept in the millisecond scale. Since searchers prefer sound to complete results, steps towards this direction should sacrifice completeness for soundness in the returned results, in order to maintain performance.

2 Inversely, end user's searches will improve. First, the amount of information returned in response to his/her queries will be more accurate. In addition, this information, being semantically rich, will allow for semantic searches in contrast to the conventional keyword-based, syntactic searches. From the Web user/searcher's perspective, such an evolution means queries closer to natural language and results that demonstrate intelligence. Consider the True Knowledge query engine (www.trueknowledge.com) for instance. While current search engines, in a query of the form "Is Jennifer Lopez single?" will capture the keyword "single" and will return tons of information about the singles the artist has released, in the True Knowledge answer engine context, the query is processed, deducing that "single" refers to marriage or relationship status, and the engine returns a simple "No".

This will radically change the way users interact with search engines: instead of mere document directories, the latter will turn into knowledge producers and recommendation applications matching user's preferences à la Hunch (`hunch.com`) or Goby (`goby.com`). From that point, possibilities are endless, and engines that will reply to a natural language sentence (e.g. "I want to go out tonight, but not to be home very late") with an appropriate recommendation suiting the user's interests, preferences, whereabouts and time schedule, seem to be just around the corner. Furthermore, the user will be able to inspect more easily the accuracy and validity of the data results, by judging from the trustworthiness of other Linked Data sources linking to the specific result and from additional provenance information.

3 Companies can benefit as well by allowing their data to be indexed by search engines. Web pages with a larger amount of content are (even intuitively) more useful. If this aspect is taken into account by search engines, higher visibility (i.e. rankings in results pages) will increase Web site traffic, the number of potential and actual customers and consecutively, higher revenues.

In the same time, more intelligent, semantic searches are allowed to be performed on the same data. For example a business publishing at its Web site, products and services using the GoodRelations ontology (Hepp, 2008), will eventually allow users to find products or services through specific queries like an offering for a TV screen with a size ranging from 30 to 40 inches, a price between 700 and 800 euros and one-day free shipping. Currently, publishing such semantic descriptions of companies and products, is in an embryonic state, due to the difficulties of the procedure and the not so obvious profits. Even if the hard work of describing product data for a company is done, there will be technical issues regarding the consumption of this data from the major search engines. For example, Yahoo SearchMonkey currently considers RDF data only if it is either submitted via the (proprietary) DataRSS feed format or if it is embedded inside XHTML pages via RDFa, meaning that an RDF/XML file describing the same data is not taken into account if not submitted directly to a Semantic Web search engine. Until the exploiting procedure is clear and straightforward and a working scenario is presented, the companies will not be willing to do the extra work.

4.2 Linked Data and Company Environments

The basic tenet behind today's Web content is mostly “Anyone can say Anything about Any topic" (AAA as it is often referred to). This means that any individual is given the freedom to express any piece of information combined with information from any other source (Allemang and Hendler, 2008a). As a result, information that can be found on the Web is not always accurate. The ultimate goal of the Semantic Web is to bring order and trust in this chaos of on-line information so as to enable the Web user to effectively, conveniently and quickly search and find accurate information. In order to turn this vision into a reality, the proposed solution is based on Linked Data, as analysed in Section 3.1.

Linked Data allows querying across data sources. In the simplest sense, a focal point is provided for referencing (referring to) and de-referencing (retrieving data about) any given Web resource. The prevailing benefits that occur for the Web user and the Web itself from this capability stem from data integration at a semantic level.

However, Linked Data usage does not target only on-line content. When talking about markets, we should distinguish between applications that rely on the public Web of Data and applications inside companies. A “behind-the-scenes” transition to semantic technologies can bring lots of benefits in enterprise applications as it allows seamless interaction with distributed heterogeneous data sources. This approach is technology-agnostic and, in addition, it allows transcendence of the conventional RDBMS models, vendor-specific APIs and Web Services. Integration of Linked Data internally in enterprise environments can bring intelligence and independence from technologies in typical company and cross-company systems (e.g. ERP, CRM, HR and Marketing systems).

First of all, adoption of semantic technologies for in-house usage in a company can ease *interoperability* among distributed software components. Current approaches regarding interoperability can be categorised as being mapping-based, intermediary-based or query-oriented (Park and Ram, 2004), all of which are perfectly suitable for semantically rich approaches.

Integration is a concept different than interoperability but relies on it. In general, the architecture of a data integration system comprises the local schemata of the sources and a global schema on which queries can be submitted. Using common ontologies to describe the local schemata and SPARQL endpoints to perform queries is an approach that can offer an intelligent alternative to current syntactic integration approaches.

Semantic Web technologies can also be used behind the scenes for *system modeling*. A model can be used as a mediator among multiple viewpoints, it can be used as an interfacing mechanism between humans or computers to understand each other, even offer useful predictions. The usage of ontologies in systems modeling provides powerful means for the achievement of an abundant system description in description logic (DL) terms. DL allows systems modeling in detail by deriving a concept hierarchy and a corresponding property hierarchy. However, the strength of the Semantic Web is not restricted in concept description. Model checking can be realised by the concurrent use of a reasoner, a practice that assures the creation of coherent, consistent models. The goal is to exploit the ontologies' inference support, the formally defined semantics, the support of rules, and logic programming in general (Kappel *et al.*, 2006).

Despite the fact that interoperability, integration and modeling are crucial tasks for the development of sophisticated software systems that can serve inhouse or B2B purposes and semantic technologies constitute a powerful candidate solution, their use is not commonplace in the majority of the companies. Our plea for adoption of the semantic technologies by the driving forces of the industry, including the search engines as analysed previously, is based on this observation. The role of W3C and other standard bodies is to promote the relevant standards, most of which are mature enough as analysed thoroughly in Section 2 but still, recommendations that would boost the adoption of the relevant technologies are missing. Placed in this setting, the Web engineer's position can be awkward when interacting with the non-technical management layers. A safe approach however, would be to favour semantic technologies without sacrificing any of the functionality of the internal system and maintain backwards compatibility.

5 Summary and Conclusions

Having presented an overview of the Semantic Web landscape, the main question to which we attempted to provide an answer in this paper, is why the Semantic Web, despite its long presence, the maturity of its theoretical and (partly) technical aspects, the large enthusiastic community, has not yet established strong bonds outside academia. More specifically, we argue that ongoing research should be based on three major axes:

1 *Simplicity*, mostly entailed by automation. Current approaches in semantically annotating and publishing content allow for substantial results. What is lacking though is automation: for instance, it is not possible to manually annotate bulks of information. As discussed in Section 3.2, current annotation capabilities offered to Web engineers produce static results in the sense that it is impossible for one person or even a team to annotate several thousands of pages that can be auto-generated from a database backend. In addition, annotation is highly prone to human errors, it can be easily outdated and needs effort to keep it consistent. Automation in creation and maintenance is needed to reduce the amount of resources in time and money a company needs to publish semantically rich content or to enrich and add intelligence to its internal software subsystems.

In the same manner, automation needs to be offered to the end user and more importantly to the Web engineer, regarding the semantic annotation of the published content. Despite its importance, semantic annotation is not always present. It is a time-consuming task and users do not usually consider it important enough to spend time annotating the already published content. The companies on the other hand mostly believe that annotation is a burden in resources in terms of time and money. Moreover, the reuse of this information is troublesome as annotation is usually likely to be redundant, partial or stored in different formats (Iria *et al.*, 2004). If we add to the above the fact that annotation easily becomes outdated, then we can easily state that the commercial future of the Semantic Web is endangered (Uren *et al.*, 2006). Thus, automation is a crucial requirement that needs to be addressed.

2 *Integration with existing technologies* without sacrificing established technologies (e.g. regarding security and performance). Semantic technologies do not offer a substitute for current practices; they rather complement them. The Web engineer need not abandon his/her experience but instead, build on top of it. Experience in technologies involving information administration and processing should constitute the basis for further developments. Legacy systems need not be substituted. It is obvious that none of the existing characteristics, for instance speed, should be sacrificed. The Semantic Web needs to co-exist with established practices, technologies and add to existing

information, not modify it. This has always been its goal: to model knowledge and add semantics to it.

3 *Adoption by the driving forces of the Web industry*. This seems to be the most promising solution for the chicken-and-egg problem of the Semantic Web. Strong incentives will be given to the companies, and great benefits for the end user. Search engines for instance, as analysed in Section 4.1 can offer more accurate results to the end users/searchers by taking into account semantic annotations and thus derive great benefits for the services offered. Actions towards this direction will act as a catalyst for further adoption of semantic technologies, since Web content publishers are always interested in search engine visibility.

However, as there are always two sides in the same coin, several of the benefits the Semantic Web evangelises cannot be realised unless some sacrifices are made. Such trade-offs have to be considered and often, the decision of the amount of qualities to be lost or gained is a crucial one. Below, we briefly sketch only few of these challenges:

1 *Automation vs. soundness/completeness.* The more automated a method is, the fewer the correct results in its output are. Considering the annotation task, which ideally requires some human intervention, the above statement means that there is no fully automated method that annotates correctly all the components of the resource under examination (be it a Web document, a database, or an image). In such annotation problems, there are usually cases whose semantics are not distinguishable by any algorithm. Thus, a decision has to be made considering the amount of automation with regard to the desired soundness and completeness of the result.

2 *Adoption of semantic technologies vs. re-engineering of current practices.* While it is well expected that search engine and other business companies that choose to employ semantic technologies will gain several advantages (as described in Section 4), the transition cost in terms of systems' modifications and business process re-engineering for these companies seems often unbearable. Smart but not radical decisions need to be made for smooth business transformation.

3 *Expressiveness vs. complexity.* It is well known that the more expressive a knowledge description language is, the more complex the reasoning task is. Increased expressiveness allows for more accurate modeling and provides the ability to better capture real-world domain semantics perceived by the user, but accounts as well for increased reasoning complexity and response time, which must be kept low for acceptable quality of service.

Finally, let us analyse the Semantic Web landscape regarding the key players that contribute to its shaping. The following list is twofold: it sums up our key ideas regarding what the Semantic Web has to offer to each player if the proposed ideas gain wider acceptance, and also what is required by each one of them in order to advance towards these directions.

1 *Web Users.* They can benefit by the more intelligent queries they can pose to search engines or even to smaller pools of information. They can also attain better on-line presence. Web users, being passive recipients of technological evolutions, are not required to do anything to promote the Semantic Web vision. Nevertheless, Web users can be regarded as generators of valuable semantic content, mainly by tagging multimedia resources, the semantics of which are often implicit and troublesome to extract. From the latter point of view, it can be argued that finding ways to motivate casual Web users to perform semantic annotation can be the key for the widespread adoption of the Semantic Web. One way to achieve this is the development of user-friendly tools and applications that ease semantic annotation and present graphically the result of the user's action that, more or less, adds to the overall machine understandable knowledge. It is possible that such applications will target the natural inclination of some users to “fill the gaps” of incomplete knowledge, thus encouraging them to generate large quantities of semantic content.

2 *Web Engineers.* In the hands of a Web engineer, semantic technologies offer intelligence. Practically, they can complement information integration, interoperability and in general, management efforts and lead to more effective coding, for instance in publishing Web content, rules definition, remote code execution, and modeling.

It is required from their part to make choices without affecting core business procedures. A co-existence with current deployments, that can be automatically updated, is needed. A large degree of automation and metadata auto-generation is needed while in the same time maintaining consistency and backwards compatibility.

3 *Search Engines.* Their role is crucial since they constitute the main gateway to access information. Semantic technologies can allow for indexing the Deep Web, in contrast with the current Surface Web indexing. Consecutively, this can provide end user searches with more accurate results.

What is required is to retain their current properties in terms of simplicity and performance. What would be desirable is to favour the use of microformats or semantic markups, which can be materialised by offering higher ranks or merely more “attractive results”, a practice that would give strong motivation for publishing semantically annotated content by companies.

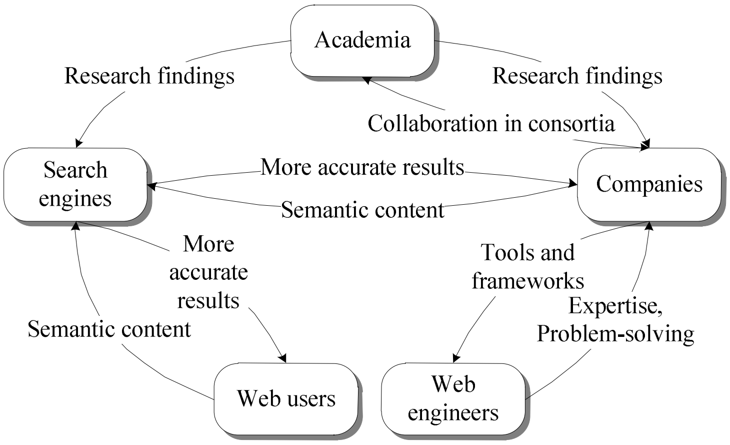
4 *Companies.* Most importantly, the Semantic Web can offer behind-the-scenes enhancements in internal systems and add intelligence in interoperability and integration among distributed systems, and be used to cover modeling needs. Combined with the role of the search engines, it can potentially offer higher visibility in semantic search engines.

What is required by the companies, mainly by tool and framework manufacturers, is the production of automated frameworks and code libraries that will exploit semantic capabilities and integrate them in the functionality offered in their products.

5 *Academia.* Basic research into the core concepts and fundamental technologies that constitute the Semantic Web landscape has reached maturity. What is missing from the big picture is research in automation that can allow simplicity to be integrated in semantic problem-solving approaches. Also, still research needs to be conducted in crucial matters such as security and privacy.

The interactions among these key players are illustrated in Figure 1 that sums up our observations and recommendations.

Figure 1 Interactions in the Semantic Web environment



In this paper, we analysed the main reasons that preserve the Semantic Web bottleneck and we offered a point of view over the actions that need to be taken from each constituent entity towards solving each problem. The main conclusion that can be drawn by observing the general picture is that the first steps have been made. Even though semantically-enabled systems are absent from the end users' everyday Web experience, these systems are ready to be released "into the wild". The most interesting observation from (Cardoso, 2007) is that the respondents asked to estimate how long it would be before they put their ontology-based system into production, replied that they are already in the process of developing and installing such a system (25,44 percent), they plan to go into production in the next six months (20,95 percent), or they will wait for a year or more (25,69 percent). Only 27,93 percent state that they do not have such plans for the future. Even though the survey was conducted mostly in the academic world, it reveals that the time is short for the Semantic Web.

However, each player's cards need to be played appropriately. Much still needs to be done in order to effectively publish and exploit large-scale semantic information. Following the approach suggested in this paper, we are confident that the Semantic Web bottleneck will be shortly circumvented and the Semantic Web vision will be at last realised.

Acknowledgements

Dimitrios-Emmanuel Spanos wishes to acknowledge the financial support of 'Alexander S. Onassis Public Benefit Foundation', under its Scholarships Programme.

References

Adida, B. (2008) 'hGRDDL: Bridging microformats and RDFa', *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 6, No. 1, pp.54–60.

Alexander, K., Cyganiak, R., Hausenblas, M. and Zhao, J. (2009), 'Describing Linked Datasets - On the Design and Usage of voiD, the 'Vocabulary of Interlinked Datasets'', *Proceedings of the 2nd Workshop on Linked Data on the Web (LDOW2009).*

Allemang, D. and Hendler, J. (2008a), *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*, Chapter 1: What is the Semantic Web?, pp.1–15, Morgan Kaufmann.

Allemang, D. and Hendler, J. (2008b), *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*, Chapter 7: RDFS-Plus, pp.123–157, Morgan Kaufmann.

Ankolekar, A., Krötzsch, M., Tran, T. and Vrandečić, D. (2008), 'The Two Cultures: Mashing up Web 2.0 and the Semantic Web', *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 6, No. 1, pp.70–75.

Auer, S., Dietzold, S., Lehmann, J., Hellmann, S. and Aumueller, D. (2009), 'Triplify - light-weight linked data publication from relational databases', *Proceedings of the 18th International World Wide Web Conference*, pp.621-630.

Baeza-Yates, R., Ciaramita, M., Mika, P. and Zaragoza, H. (2008), 'Towards semantic search', *Proceedings of the 13th international conference on Natural Language and Information Systems (NLDB'08)*, pp.4–11, Berlin, Heidelberg, Springer-Verlag.

Becker, C. and Bizer, C. (2008), 'DBpedia mobile: a location-enabled linked data browser', *Proceedings of the 1st International Workshop on Linked Data on the Web*, Beijing, China.

Belhajjame, K., Embury, S. M., Paton, N. W., Stevens, R. and Goble, C. A. (2008), 'Automatic annotation of Web services based on workflow definitions', *ACM Transactions on the Web*, Vol. 2, No. 2, pp.1–34.

Bergman, M. K. (2001), 'The Deep Web: Surfacing Hidden Value' *Journal of Electronic Publishing*, Vol. 7, No. 1, pp.1–17.

Berners-Lee, T., Hendler, J. and Lassila, O. (2001), 'The Semantic Web – A New Form of Web Content that is Meaningful to Computers Will Unleash a Revolution of New Possibilities', *Scientific American*, Vol. 284, No. 5, pp.34–43.

Bizer, C., Heath, T., Idehen, K. and Berners-Lee, T. (2008), 'Linked Data on the Web', *Proceedings of the 17th International Conference on World Wide Web (WWW'08)*, pp.1265–1266, New York, NY, USA, ACM.

Bizer, C. and Seaborne, A. (2004), 'D2RQ – Treating Non-RDF Databases as Virtual RDF Graphs', *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004)*.

Blakeley, C. (2007), 'Mapping Relational Data to RDF with Virtuoso's RDF Views', White paper, OpenLink Software, available online at `http://virtuoso.openlinksw.com/Whitepapers/html/rdf_views/virtuoso_rdf_views_example.html` (accessed April 2010).

Brickley, D. and Guha, R. (1999), *Resource Description Framework (RDF) Schema Specification*, available online at `http://www.w3.org/TR/1999/PR-rdf-schema-19990303/` (accessed April 2010).

Broekstra, J., Kampman, A. and van Harmelen, F. (2003), *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, Chapter 7: Sesame: An Architecture for Storing and Querying RDF Data and Schema Information, pp.197–222, MIT Press.

Cardoso, J. (2007), 'The Semantic Web Vision: Where are We?', *IEEE Intelligent Systems*, Vol. 22, No. 5, pp.84–88.

Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A. and Wilkinson, K. (2004). 'Jena: Implementing the Semantic Web Recommendations', *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pp.74–83, ACM New York, NY, USA.

Cimiano, P., Haase, P., Heizmann, J., Mantel, M. and Studer, R. (2008), 'Towards portable natural language interfaces to knowledge bases - The case of the ORAKEL system', *Data and Knowledge Engineering*, Vol. 65, No. 2, pp.325–354.

Connolly, D. (2007), *Gleaning Resource Descriptions from Dialects of Languages (GRDDL)*, available online at `http://www.w3.org/TR/grddl/` (accessed April 2010).

d'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V. and Guidi, D. (2008), T'oward a New Generation of Semantic Web Applications', *IEEE Intelligent Systems*, Vol. 23, No. 3, pp.20–28.

Doorenbos, R. (1995), *Production Matching for Large Learning Systems*, PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

Dubost, K. and Herman, I. (2008), *The state of the Semantic Web*, presentation in the INTAP Semantic Web Conference, Tokyo, Japan, slides available online at `www.w3.org/2008/Talks/0307-Tokyo-IH/Slides.pdf` (accessed April 2010).

Edwards, J. and DeVoe, D. (1997), *3-tier client/server at work*, John Wiley & Sons, Inc., New York, NY, USA.

Feigenbaum, E. (1977), 'The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering', *Proceedings of the 5th International Joint Conference on Artificial Intelligence 1977*, pp.1014–1029, William Kaufmann.

Francois, A. R. J., Nevatia, R., Hobbs, J. and Bolles, R. C. (2005), 'VERL: An ontology framework for representing and annotating video events', *IEEE Multimedia*, Vol. 12, No. 4, pp.76–86.

Gruber, T. (2008), 'Collective knowledge systems: Where the Social Web meets the Semantic Web', *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 6, No. 1, pp.4–13.

Halevy, A., Norvig, P. and Pereira, F. (2009), 'The Unreasonable Effectiveness of Data', *IEEE Intelligent Systems*, Vol. 24, No. 2, pp.8–12.

Hallmark, G., de Sainte Marie, C., Fabro, M. D., Albert, P. and Paschke, A. (2008), 'Please Pass the Rules: A Rule Interchange Demonstration' *Proceedings of the International Symposium on Rule Representation, Interchange and Reasoning on the Web*, pp.227–235, Springer.

Hausenblas, M. (2009). 'Linked Data Applications – The Genesis and the Challenges of Using Linked Data on the Web', Technical Report, DERI Galway, available online at `http://linkeddata.deri.ie/sites/linkeddata.deri.ie/files/lod-app-tr-2009-07-26_0.pdf`, (accessed April 2010).

He, B., Patel, M., Zhang, Z., and Chang, K. (2007), 'Accessing the Deep Web', *Communications of the ACM*, Vol. 50, No. 5, pp.94–101.

Hendler, J. (2008), 'Web 3.0: Chicken Farms on the Semantic Web', *IEEE Computer*, Vol. 41, No. 1, pp.106–108.

Hepp, M. (2008), 'GoodRelations: An Ontology for Describing Products and Services Offers on the Web', *Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW2008)*, pp.329–346.

Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B. and Dean, M. (2004), *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, World Wide Web Consortium, Member Submission, available online at `http://www.w3.org/Submission/SWRL/` (accessed April 2010).

Horrocks, I., Patel-Schneider, P. and van Harmelen, F. (2003), 'From SHIQ and RDF to OWL: The Making of a Web Ontology Language', *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 1, No. 1, pp.7–26.

Iqbal, A., Ureche, O., Hausenblas, M. and Tummarello, G. (2009), 'LD2SD: Linked Data Driven Software Development', *Proceedings of the 21st International Conference on Software Engineering and Knowledge Engineering (SEKE'09)*, pp.240–245.

Iria, J., Ciravegna, F., Cimiano, P., Lavelli, A., Motta, E., Gilardoni, L. and Mönch, E. (2004), 'Integrating Information Extraction, Ontology Learning and Semantic Browsing into Organizational Knowledge Processes', *Proceedings of the EKAW Workshop on the Application of Language and Semantic Technologies to support Knowledge Management Processes, at the 14th International Conference on Knowledge Engineering and Knowledge Management*.

Kappel, G., Kapsammer, E., Kargl, H., Kramler, G., Reiter, T., Retschitzegger, W., Schwinger, W. and Wimmer, M. (2006), 'Lifting Metamodels to Ontologies: A Step to the Semantic Integration of Modeling Languages', *Proceedings of Model Driven Engineering Languages and Systems (MoDELS'06)*, pp.528–542, Springer.

Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D. and Scholl, M. (2002), 'RQL: A Declarative Query Language for RDF', *Proceedings of the 11th International World Wide Web Conference (WWW'02)*, pp.592-603.

Khare, R. (2006), 'Microformats: the next (small) thing on the Semantic Web?', *Internet Computing, IEEE*, Vol. 10, No. 1, pp.68–75.

Kifer, M. (2008), 'Rule Interchange Format: The Framework' *Rule Representation, Interchange and Reasoning on the Web: International Symposium, RuleML*, pp.1–11, Springer-Verlag Inc., New York, NY, USA.

Konstantinou, N., Spanos, D.E. and Mitrou, N. (2008), 'Ontology and Database Mapping: A Survey of Current Implementations and Future Directions', *Journal of Web Engineering*, Vol. 7, No. 1, pp.1–24.

Konstantinou, N., Solidakis, E., Zafeiropoulos, A., Stathopoulos, P. and Mitrou, N. (2010), 'A Context-aware Middleware for Real-Time Semantic Enrichment of Distributed Multimedia Metadata', *International Journal of Multimedia Tools and Applications*, Vol. 46, No. 2, pp.425-461.

Lopez, X. and Das, S. (2007), 'Semantic Data Integration for the Enterprise', White paper, Oracle Corporation, available online at http://www.oracle.com/technology/tech/semantic_technologies/pdf/semantic11g_dataint_twp.pdf (accessed April 2010).

McGuinness, D. and van Harmelen, F. (2003), *OWL Web Ontology Language Overview*, available online at http://www.w3.org/TR/2003/PR-owl-features-20031215/ (accessed April 2010).

Mika, P. (2007), 'Ontologies are us: A unified model of social networks and semantics', *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 5, No. 1, pp.5–15.

Mika, P. (2008), 'Microsearch: An Interface for Semantic Search', *Proceedings of the Workshop on Semantic Search (SemSearch 2008) at the 5th European Semantic Web Conference (ESWC 2008)*, Tenerife, Spain.

Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A. and Lutz, C. (2009), *OWL 2 Web Ontology Language Profiles*, available online at http://www.w3.org/TR/2009/PR-owl2-profiles-20090922/ (accessed April 2010).

Oren, E., Kotoulas, S., Anadiotis, G., Siebes, R., ten Teije, A. and van Harmelen, F. (2009), 'Marvin: Distributed reasoning over large-scale Semantic Web data', *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 7, No. 4, pp.305-316.

Papamarkos, G., Poulovassilis, A. and Wood, P. T. (2003), 'Event-Condition-Action Rule Languages for the Semantic Web', *Workshop on Semantic Web and Databases (SWDB 03)*, pp.309–327.

Park, J. and Ram, S. (2004), 'Information Systems Interoperability: What Lies Beneath?', *ACM Transactions on Information Systems*, Vol. 22, No. 4, pp.595–632.

Passant, A. and Laublet, P. (2008), 'Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data', *Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008)*, Beijing, China.

Perez, J., Arenas, M. and Gutierrez, C. (2006), 'Semantics and Complexity of SPARQL', *International Semantic Web Conference 2006 (ISWC 2006)*, pp.30-43.

Petridis, K., Anastasopoulos, D., Saathoff, C., Timmermann, N., Kompatsiaris, I. and Staab, S. (2006), 'M-OntoMat-Annotizer: Image Annotation Linking Ontologies and Multimedia Low-Level Features', *Engineered Applications of Semantic Web Session (SWEA) at the 10th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES'06)*, Bournemouth, UK, pp.633-640.

Polleres, A. (2007), 'From SPARQL to rules (and back)' *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, pp.787–796, New York, NY, USA, ACM Press.

Prud'hommeaux, E. and Seaborne, A. (2007), *SPARQL Query Language for RDF*, available online at http://www.w3.org/TR/2007/PR-rdf-sparql-query-20071112/ (accessed April 2010).

Schenk, S. (2007), 'A SPARQL semantics based on Datalog', *Proceedings of the 30th annual German conference on Advances in Artificial* Intelligence, pp.160–174.

Schroeter, R., Hunter, J., Guerin, J., Khan, I. and Henderson, M. (2006), 'A Synchronous Multimedia Annotation System for Secure Collaboratories', *Proceedings of the Second IEEE International Conference on e-Science and Grid Computing (E-SCIENCE'06)*, pp.41-48, Washington, DC, USA, IEEE Computer Society.

Shadbolt, N., Berners-Lee, T. and Hall, W. (2006), 'The Semantic Web Revisited' *IEEE Intelligent Systems*, Vol. 21, No. 3, pp.96–101.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A. and Katz, Y. (2007), 'Pellet: A Practical OWL-DL Reasoner', *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 5, No. 2, pp.51–53.

Ureche, O., Iqbal, A., Cyganiak, R. and Hausenblas, M. (2009), 'Accessing Site-Specific APIs

Through Write-Wrappers From The Web of Data', *Proceedings of the Semantics for the Rest of Us Workshop (SemRUs) at ISWC 2009*, Washington DC, USA, pp.1–11.

Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E. and Ciravegna, F. (2006), 'Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art', *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 4, No. 1, pp.14–28.

Weiske, C. and Auer, S. (2007), 'Implementing SPARQL Support for Relational Databases and Possible Enhancements', *Proceedings of the 1st Conference on Social Semantic Web* (*CSSW 2007)*, Leipzig, Germany, pp.69–80.

Wilde, E. and Hausenblas, M. (2009). 'RESTful SPARQL? You name it!: aligning SPARQL with REST and resource orientation', *Proceedings of the 4th Workshop on Emerging Web Services Technology*, pp. 39–43, ACM.

Zhang, J. and Dimitroff, A. (2005a), 'The impact of webpage content characteristics on webpage visibility in search engine results (Part I)', *Information Processing and Management*, Vol. 41, No. 3, pp.665–690.

Zhang, J. and Dimitroff, A. (2005b), 'The impact of metadata implementation on webpage visibility in search engine results (Part II)' *Information Processing and Management*, Vol. 41, No. 3, pp.691–715.

Zhao, J., Miles, A., Klyne, G. and Shotton, D. (2009), 'Linked data and provenance in biological data webs', *Briefings in bioinformatics*, Vol. 10, pp.139–152.

Notes

1 Research conducted while Nikolaos Konstantinou was with the National Technical University of Athens.