Toward a New Generation of Semantic Web Applications

Mathieu d'Aquin, Enrico Motta, Marta Sabou, Sofia Angeletou, Laurian Gridinoc, Vanessa Lopez, and Davide Guidi, *Open University*

A new generation of applications offers insight into the Semantic Web's current and future challenges—as well as the opportunities it might provide for users and developers alike.

Although research on integrating semantics with the Web started almost as soon as the Web was in place, a concrete Semantic Web—that is, a large-scale collection of distributed semantic metadata—emerged only over the past four to five years. The Semantic Web's embryonic nature is reflected in its existing applications. Most of these applications tend to produce and consume their own data, much like traditional knowledge-based applications, rather than actually exploiting the Semantic Web as a large-scale information source.[1]

These first-generation Semantic Web applications[1] typically use a single ontology that supports integration of resources selected at design time. An early influential example from the academic world is CS Aktive Space (http://cs.aktivespace.org). This application combines data about UK computer science research from multiple, heterogeneous sources (such as databases, Web pages, and RDF data) and lets users explore the data through an interactive portal. Not surprisingly, this paradigm also informs recently launched commercial solutions based on Semantic Web technology. For example, Garlik.com's personal-information-management service uses ontologies to discover and integrate personal financial data from the Web. Similarly, corporate Semantic Webs—which Gartner Consulting highlighted in 2006 as a key strategic technology trend—use a corporate ontology to drive the semantic annotation of organizational data and thus facilitate data retrieval, integration, and processing. Corporate Semantic Web application areas include the car industry (such as Renault's system for managing project history), the aeronautical industry (such as Boeing's use of semantic technologies to gather corporate information), and the telecommunication industry (such as British Telecom's system for enhancing digital libraries).

Although corporate Semantic Webs 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. To address this, we began an ambitious research program two years ago dubbed "Next-Generation Semantic Web Applications." Our project's objective was to experiment with a new class of applications that would go beyond classic corporate Semantic Webs and intelligently exploit the Semantic Web as a large-scale, heterogeneous semantic resource. Our research also highlighted some key achievements so far, as well as several obstacles that must be tackled if we're to realize the vision of the Semantic Web as a large-scale enabling infrastructure for both data integration and a new generation of intelligent applications.

1541-1672/08/$25.00 © 2008 IEEE
Published by the IEEE Computer Society

AI and the Semantic Web

Although much early AI research focused on general methods for problem solving and efficient theorem proving, by the mid-1970s, many AI researchers realized this essential point:

> The fundamental problem of understanding intelligence is not the identification of a few powerful techniques, but rather the question of how to represent large amounts of knowledge in a fashion that permits their effective use.[2]

Accordingly, these researchers advocated a paradigm shift, moving away from "weak" reasoning and problem-solving techniques and toward the creation of effective methods for acquiring, representing, and reasoning with large amounts of domain knowledge. A few years later, Brian Smith precisely formulated this knowledge-based paradigm when he defined the knowledge-representation hypothesis:

> Any mechanically embodied intelligent process will be comprised of structural ingredients that we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and independent of such external semantic attribution, play a formal but causal and essential role in engendering the behaviour that manifests that knowledge.[3]

Hence, the essential element of AI's knowledge-based paradigm is this causal relationship between a system's explicit knowledge representation and its (intelligent) behavior. Unfortunately, the paradigm has a key problem in its so-called *knowledge acquisition bottleneck*.[4] This KA bottleneck concerns the difficulty of acquiring, representing, and maintaining an intelligent system's knowledge base.

Revisiting the KA bottleneck

Although many people (especially those critical of AI in general) focus on the KA bottleneck's epistemological aspects—that is, the difficulty inherent in formalizing expertise for computer processing—in practice, the issue tends to be primarily economic. If a knowledge-based system (KBS) is to be economically feasible, the cost of acquiring and maintaining its knowledge base must be significantly less than the economic benefits derived from the system's deployment. Hence, pragmatically, the KA bottleneck simply means that it's often too expensive to acquire and encode the large amount of knowledge that an application needs.

For these reasons, much of the key KBS research of the past 20 years has tackled the KA bottleneck and developed methods for knowledge sharing and reuse. The goal was to make the knowledge-engineering process more robust and cost-effective. This line of research has produced the key AI technologies for specifying reusable model components (ontologies)[5] and reasoning components (problem-solving methods)[6,7] and clearly bears a direct impact on current Semantic Web technologies. Specifically, ontologies provide the core technology for the Semantic Web's data interoperability, while emerging standards for Semantic Web Services, such as the Web Service Modeling Ontology (www.wsmo.org), inherit their conceptual foundations from research in problem-solving methods.[8]

By providing the means for large-scale distributed knowledge publishing and access, the Semantic Web could open the way to a new generation of intelligent applications.

Despite its strong AI research connection, the Semantic Web isn't AI—as its key advocates, such as Tim Berners-Lee, often emphasize. AI is about engineering intelligent machines; the Semantic Web is a technological infrastructure to enable large-scale data interoperability (the so-called "Web of data"). Although this distinction is important, there's another interesting hypothesis here. In addition to providing an infrastructure for large-scale publication, integration, and reuse of semantically characterized information—much like the network of semiautomated knowledge services that Mark Stefik called "the new knowledge medium" in his extraordinarily visionary 1986 paper[9]—the Semantic Web could also provide a new context in which to address the KA bottleneck. Specifically, by providing the means for large-scale distributed knowledge publishing and access, the Semantic Web could open the way to a new generation of intelligent applications that go beyond the closed domains of traditional KBSs and exploit semantic information on a large scale. (By "traditional KBS," we mean a computer system that relies, on one hand, on the knowledge formalized in a knowledge representation language and, on the other hand, on reasoning mechanisms for problem solving.)

Semantic Web applications vs. the traditional KBS

Although the vision of powerful, nonbrittle intelligent systems is appealing, moving from the classic KBS to the Semantic Web implies a dramatic shift in context. Early attempts at tackling the KA bottleneck, such as Cyc (www.cyc.com), did so by creating a very large, high-quality knowledge base. However, if we view the Semantic Web as a very large knowledge base, several key differences from classic KBSs become apparent:

- *Heterogeneity.* Typically, developers construct knowledge bases according to (at most) a few small sets of carefully designed and integrated ontologies. The Semantic Web is characterized by heterogeneity along several dimensions, such as ontology encoding, quality, complexity, modeling, and views. Hence, an application using data from multiple sources involves a nontrivial integration effort.
- *Quality.* To ensure quality, developers build classic knowledge bases in a centralized fashion, typically using a small team of knowledge engineers. As a result, trust isn't an issue. On the Semantic Web, information originates from many different sources and varies considerably in quality. Trust is therefore a key issue on the Semantic Web.
- *Scale.* With its millions of documents and billions of triples, the Semantic Web is already well beyond the size of a classic KBS. Although applications typically focus on specific Semantic Web subsets, efficient access and information processing nonetheless require a quantum leap in applications' ability to locate and process relevant information.
- *Reasoning.* Traditional KBSs derive their power from sophisticated reasoning mechanisms that combine high-quality knowledge bases with powerful models of

generic tasks such as planning, diagnosis, and scheduling.

Regarding the last distinction, because the Semantic Web combines heterogeneity, variable data quality, and scale, the applications we envision will exhibit intelligent behavior owing less to an ability to carry out complex inferencing than an ability to exploit the large amounts of available data. That is, as we move from classic KBSs to Semantic Web applications, intelligence becomes a side effect of scale, rather than of sophisticated logical reasoning. An important corollary here is that, as logical reasoning becomes less important and scale and data integration become key issues, other types of reasoning—based on machine learning, linguistic, or statistical techniques—become crucial, especially because they frequently need to integrate and use other, nonsemantic data. Indeed, as we describe later, all our applications integrate different forms of reasoning.

Although the hypothesis of using the Semantic Web as a large-scale knowledge source opens up many exciting opportunities, to realize it in practice, we must design applications that are quite different from classic KBSs. Such next-generation Semantic Web applications must address significant problems associated with the Semantic Web's scale and heterogeneity as well as with the widely varying quality of the information it contains.

Next-generation Semantic Web applications

Our research on next-generation Semantic Web applications originates from our observation—and anticipation—that intelligent-application development will increasingly change owing to the availability of the Semantic Web's large-scale, distributed body of knowledge.[1] Dynamically exploiting this knowledge introduces new possibilities and challenges requiring novel infrastructures to support the implementation of next-generation Semantic Web applications.

Key features and requirements

Next-generation Semantic Web applications achieve their tasks by automatically retrieving and exploiting knowledge from the Semantic Web as a whole. Unlike early Semantic Web applications, which gathered and engineered knowledge at design time, these new applications explore the Web to discover ontologies relevant to the task at hand. Because dynamic knowledge reuse replaces the traditional knowledge-acquisition task, we can potentially reduce the application development cost. In addition, because such applications can use any semantic information available online, they're not necessarily bound to a particular domain.

Still, as we discussed earlier, next-generation Semantic Web applications face novel challenges related to scale, heterogeneity, and information quality. To tackle these challenges, the applications require new mechanisms and tools that aren't needed in classic KBSs because their knowledge is manually selected and integrated.

Any application that wishes to explore large-scale semantics must perform the following tasks:

> Watson provides efficient services to support application developers in exploiting the Semantic Web's voluminous distributed and heterogeneous data.

- *Find relevant sources.* The ability to dynamically locate sources with relevant semantic information is a prerequisite for applications that aim to leverage online knowledge. This feature is important because developers might not be able to judge a particular resource's relevance to the target problem at design time.
- *Select appropriate knowledge.* Applications must select the appropriate knowledge from the set of previously located semantic documents on the basis of application-dependent criteria, such as data quality and adequacy to the task at hand.
- *Exploit heterogeneous knowledge sources.* When reusing online semantic information, the application can't make assumptions about the ontological nature of the target elements. Hence, the process must be generic enough to use any online semantic resource. As with the two previous tasks, the application must carry out this activity at runtime.
- *Combine ontologies and resources.* Developers can't expect one unique knowledge source to provide all the required elements for a given application. Therefore, a typical next-generation Semantic Web application must select and integrate partial knowledge fragments from different sources and jointly exploit them.

Although the envisaged applications must perform these tasks to leverage online semantics, actually implementing the required mechanisms within individual applications is infeasible. What we need is a single access point that applications can reference to obtain the appropriate semantic resources. We can realize this through an infrastructure that collects, analyzes, and indexes online resources and thereby provides efficient services to support their exploitation—that is, a gateway to the Semantic Web. In principle, such a tool plays the same role as a standard Web search engine. However, in this case, the focus is on enabling semantic applications to use online knowledge.

The idea of providing efficient and easy access to the Semantic Web isn't new. Indeed, several research efforts have either considered the task as a whole or concentrated on some of its subissues. The most influential example is probably Swoogle (http://swoogle.umbc.edu), a search engine that crawls and indexes online Semantic Web documents. Swoogle claims to adopt a Web view on the Semantic Web, and, indeed, most of its techniques are inspired by traditional Web search engines. Relying on such well-studied techniques offers a range of advantages, but it also has a major limitation: by largely ignoring the semantic particularities of the indexed data, Swoogle falls short of offering the functionalities required from a truly Semantic Web gateway. Other recent Semantic Web search engines—such as Sindice (http://sindice.com) and Falcon-S (http://iws.seu.edu.cn/services/falcons/objectsearch/index.jsp)—adopt a viewpoint similar to Swoogle's and therefore suffer from the same limitations:

- *They provide only weak access to semantic information*, because they don't consider the accessed document's semantic content. Swoogle essentially treats semantic resources in the same way that Google treats Web documents. For every retrieved

ontology, for example, Swoogle displays only a text snippet showing that the queried terms occur somewhere in the ontology. The user (or application) is then supposed to download the ontology to access its content. For a human user searching the Semantic Web, this mechanism might be sufficient (although a bit inefficient); it certainly can't support semantic applications, which must be able to efficiently locate and access relevant semantic information.

- *They don't consider the quality of the knowledge they collect.* Among the Semantic Web search engines mentioned earlier, only Swoogle employs a quality criterion—specifically, a PageRank-like algorithm that provides information about a resource's "popularity." This is insufficient to support applications in assessing a semantic document's information quality and adequacy.
- *They typically pay limited attention to semantic relations between ontologies.* Swoogle, for example, considers only those relations that are explicitly stated (such as import). This is a serious limitation; as semantic resources, ontologies can be compared and related to each other through semantic relations (they might, for example, be versions of each other, mutually incompatible, and so on). This is particularly important for semantic applications that must exploit several, interrelated ontologies. In looking at results from existing Semantic Web search engines, it appears that they don't consider even the simplest (syntactic) notion of duplication (or copy), because the same documents often appear, at different ranks, several times in the results.

Watson: A Semantic Web gateway

Motivated by the needs of next-generation applications, we developed the Watson Semantic Web gateway (http://watson.kmi.open.ac.uk). Watson offers a single access point to online semantic information and provides efficient services to support application developers in exploiting this voluminous distributed and heterogeneous data. Although superficially similar to existing Semantic Web search engines, Watson overcomes their limitations by providing support for finding, selecting, exploiting, and combining online semantic resources.

To collect online semantic documents, Watson uses a set of crawlers that explore various sources, including PingTheSemanticWeb.com. Unlike standard Web crawlers, our crawlers consider both classical hyperlinks and semantic relations across documents. Also, when collecting online semantic content, they check for duplicates, copies, or prior versions of the discovered documents.

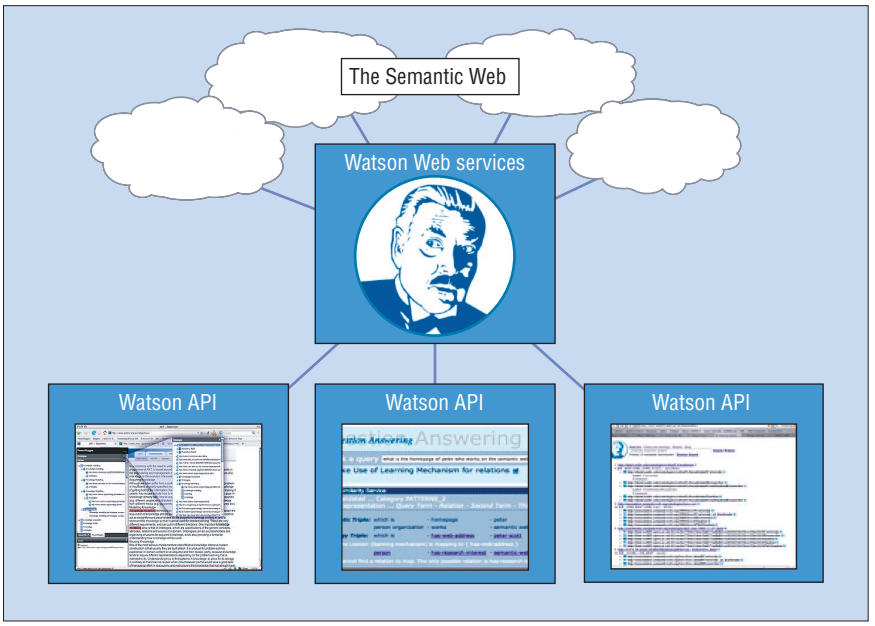

Figure 1. A Watson-based architecture for next-generation Semantic Web applications. Developers can use the Watson API to build lightweight applications, relying on the Watson gateway to exploit the knowledge available on the Semantic Web.

Once documents are collected, Watson analyzes and indexes them according to various information about each document's content, complexity, quality, and relation to other resources. This analysis step is crucial; it ensures that Watson extracts the key information, which in turn helps applications select, assess, exploit, and combine these resources.

Watson's goal is to provide applications—and, to some extent, human users—with efficient and adequate access to the information it collects. A Web interface lets users search semantic content by keyword as well as inspect and explore semantic documents. Users can also query documents using the SPARQL Protocol and RDF Query Language. However, Watson's strength is in providing the services and API needed to support the development of next-generation Semantic Web applications (see figure 1). Indeed, Watson deploys several Web services and a corresponding API that let applications

- find Semantic Web documents through a sophisticated, keyword-based search that lets applications specify queries according to several parameters (including type of entity, level of keyword matching, and so on);
- retrieve a document's metadata such as size, language, label, and logical complexity;
- find specific entities (classes, properties, individuals) within a document;
- inspect a document's content—that is, the semantic description of its entities; and
- apply SPARQL queries to Semantic Web documents.

Watson's API provides several advantages. First, unlike Swoogle and Sindice, which limit a user's number of queries per day or the number of query results, Watson doesn't restrict the amount of data it provides through its API. In our view, any piece of information Watson collects should be made available, and we provide applications with as much information as possible. Second, our API exposes a comprehensive functionalities set that lets any application use online semantic data in a lightweight fashion without having to download the corresponding semantic documents. Watson processes and indexes a semantic document's content so that applications can access it at runtime without needing sophisticated mechanisms and large resources.

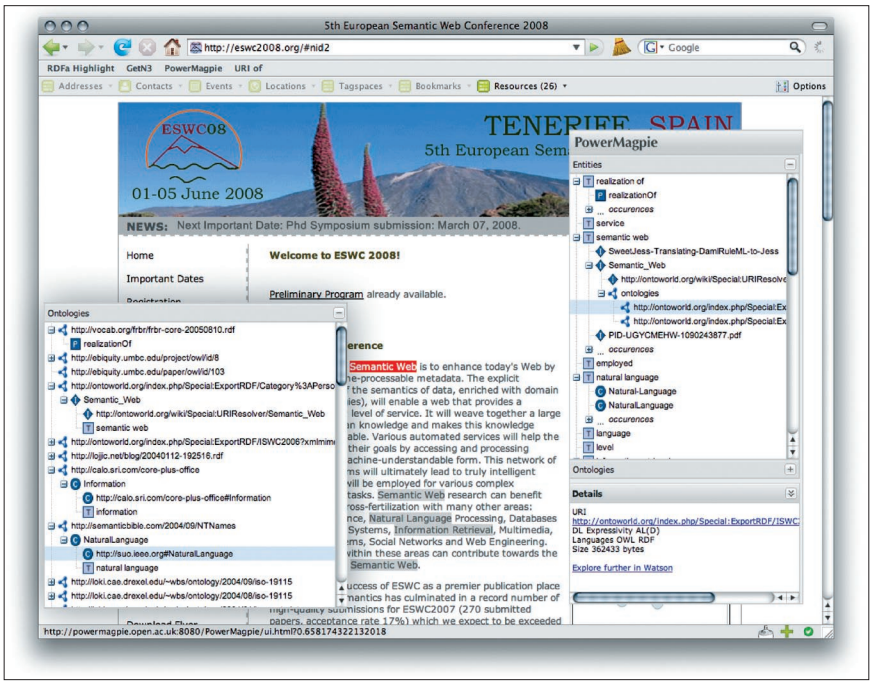

Figure 2. PowerMagpie's Entities and Ontologies panels. The Entities panel lists ontology entities that are relevant for the current Web page; the Ontologies panel shows the main ontologies that cover the Web page's text.

By providing mechanisms for searching semantic documents (keyword search), retrieving metadata about these documents, and querying their content (such as through SPARQL), Watson offers applications all the necessary elements to select and exploit online semantic resources. Moreover, the Watson Web Services and API are constantly evolving to support novel application requirements. In particular, for ranking, we're using an initial set of measures that evaluate ontology complexity and richness. We're developing a more flexible framework that combines both automatic metrics for ontology evaluation and user evaluation to allow for a more customizable selection mechanism. Another important direction is in detecting semantic relations between ontologies to support their combination. Indeed, while we have a simple duplicate detection mechanism in place, we must consider more advanced mechanisms to efficiently discover fine-grained relations, such as extension, version, or compatibility.

Exploiting large-scale semantics

Our research program was initially motivated by our development of two pioneering ontology-based applications: Aqualog (http://kmi.open.ac.uk/technologies/aqualog) for ontology-based question answering and Magpie (http://kmi.open.ac.uk/projects/magpie) for semantic browsing. Although these applications are portable from one domain to another, they subscribe to the early Semantic Web application model in that they exploit manually selected knowledge, exploring a single ontology at a time. Hence, their scope is limited by the topic domain and the selected ontology's encoded knowledge. To overcome this limitation, we envisioned their extensions—PowerAqua (http://kmi.open.ac.uk/technologies/poweraqua) and PowerMagpie (http://powermagpie.open.ac.uk)—working in an "open Web assumption," dynamically retrieving knowledge from the Semantic Web to answer questions or annotate Web pages.

Beyond PowerAqua and PowerMagpie, we're investigating this new paradigm's potential to exploit large-scale semantics through various applications, including Scarlet (http://scarlet.open.ac.uk) for ontology matching and Flor for folksonomy tagspace enrichment (defined later). Moreover, a system developed outside our research group builds on the Watson infrastructure to perform word sense disambiguation (WSD).[10]

In addition to providing concrete examples of successful next-generation Semantic Web applications, our tools and techniques offer insight into the Semantic Web's current status and its potential to support a variety of tasks.

PowerMagpie: Semantic browsing

The PowerMagpie Semantic Web browser uses openly available semantic data to help users interpret arbitrary Web page content. Unlike Magpie, which relied on a single ontology selected at design time, PowerMagpie automatically identifies and uses relevant knowledge provided by multiple online ontologies at runtime.

From a user perspective, PowerMagpie is an extension of a classic Web browser: it appears as a vertical widget at the top of browsed Web pages (see Figure 2). The widget provides several functionalities that let users explore the current Web page's semantic information. In particular, it summarizes conceptual entities relevant to the page, highlighting them in the text and letting users explore the information surrounding them in different ways. In addition, when it finds semantic information that relates the text to online semantic resources, PowerMagpie "injects" this information into the Web page as embedded annotations in RDFa. Users can then store these annotations into a local knowledge base and use them to mediate the interactions of different semantic-based systems.

Watson plays a central role in Power Magpie's architecture, providing sophisticated mechanisms for identifying and selecting ontologies relevant to the main terms extracted from a Web page. For example, unlike other search engines, Watson's ontology selection mechanism can identify a set of ontologies that jointly cover a set of terms, rather than just a single ontology that only partially covers the set of terms. Also, because the selection process relies on Watson's ontology-ranking mechanisms, it favors higher-quality ontologies.

PowerAqua: Open-domain question answering

PowerAqua's predecessor, AquaLog, derived answers to questions from a single ontology. In contrast, PowerAqua performs question answering (QA) on an unlimited number of ontologies and can automatically combine information from multiple ontologies at runtime. Users enter a question to PowerAqua in natural language; the system

then aims to return all the answers that it can find on the Semantic Web. For example, given the query, "Which are the members of the rock group Nirvana?" and two online ontologies covering the term "Nirvana"—one about spiritual stages, and one about musicians—PowerAqua can

- locate and select these two ontologies (through Watson),
- choose the appropriate ontology after disambiguating the query using the available semantic information, and
- extract an answer in the form of ontological entities.

In our example, it returns a set of individual names corresponding to the group's members: Kurt Cobain, Krist Novoselic, and Dave Grohl as well as the names of the band's earlier drummers.

We've evaluated PowerAqua's ability to derive answers from multiple ontologies selected and used on the fly during the QA process. Our evaluation showed that PowerAqua's ontology search and matching mechanisms are powerful enough to successfully map most of the questions to appropriate ontologies (see www.cisa.informatics.ed.ac.uk/OK/Deliverables/D8.5.pdf). However, our evaluation also revealed that the tool's performance was heavily influenced by the Semantic Web's data quality. For example, we submitted the query, "Which prizes have been won by Laura Linney?" Whereas the three first answers were correct, the last one was erroneous because the final ontology modeled "Laura Linney" as an instance of the class "Award." Our work was also hampered by the Semantic Web's sparseness in terms of the covered topic domains. In fact, when attempting to reuse the Text Retrieval Conference data (http://trec.nist.gov) to build our query corpus, we found that online ontologies covered only 20 percent of the topic domains described in the TREC (Text Retrieval Conference) WT10G test collection's 100 queries.

Scarlet: Relation discovery

Scarlet automatically selects and explores online ontologies to discover relations between two given concepts. When relating these concepts, Scarlet

- identifies, at runtime, online ontologies that provide information about how the two concepts relate, and
- combines this information to infer their relation.

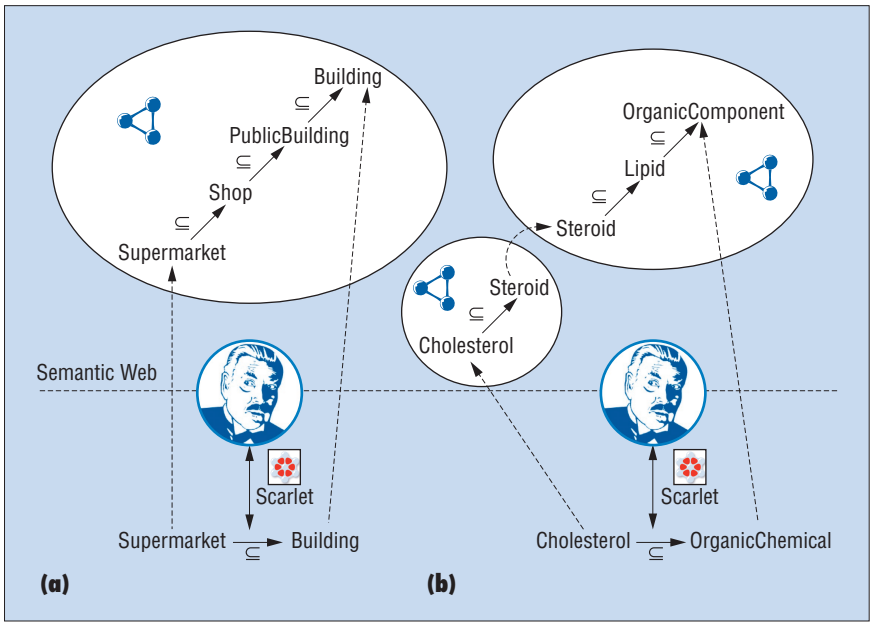

Figure 3. Scarlet's two main relation-discovery strategies. (a) Strategy S1 returns relation information defined in a single ontology. (b) Strategy S2 combines relevant information spread over two or more ontologies.

We've investigated two increasingly sophisticated strategies to discover and exploit online ontologies for relation discovery. As figure 3a shows, the first strategy, S1, derives a relation between two concepts if the relation is defined within a single online ontology—a relation between *Supermarket* and *Building* is discovered if the ontology states that *Supermarket* $\subseteq$ *Building*. In some cases, no single online ontology states the concepts' relation, as is the case with the concepts *Cholesterol* and *OrganicCompound*. To address this, the second strategy, S2, combines relevant information spread over two or more ontologies—for example, that *Cholesterol* $\subseteq$ *Steroid* in one ontology and that *Steroid* $\subseteq$ *OrganicCompound* in another (see Figure 3b). To support this functionality, Scarlet needs a Semantic Web gateway to access online ontologies. Although the first Scarlet prototype used Swoogle, its latest version leverages Watson's functionalities, which are more sophisticated. Compared to Swoogle, Watson's output contains fewer duplicate ontologies; it also ranks the ontologies it returns in terms of their semantic quality rather than their popularity. Both factors directly affect Scarlet's performance: Scarlet doesn't have to sort through redundant information, and it can typically exploit the more useful ontologies first.

We developed Scarlet on the basis of an ontology matcher that exploits Semantic Web information to discover semantic relations (mappings) between two ontologies' elements. We evaluated this matcher by aligning two large, real-life thesauri: the United Nations' 40,000-term AGROVOC thesaurus and the US National Agricultural Library's 65,000-term thesaurus.[11] Using strategy S1, we obtained a total of 6,687 mappings (2,330 subclass, 3,710 superclass, and 647 disjoint relations) by dynamically selecting, exploring, and combining 226 online ontologies. To assess the online ontologies' information quality, we manually evaluated 1,000 randomly selected mappings (about 15 percent of the alignment).

Our evaluation led us to several interesting insights about the online ontologies' quality. On the one hand, we found that the obtained mappings' precision was 70 percent and that we could raise it to 87 percent given a more sophisticated anchoring mechanism for matching terms. This finding suggests that the online ontologies' quality is good enough to produce highly precise alignments. On the other hand, our evaluation highlighted a range of typical ontology errors that can cause false mappings. One of the most common errors was the incorrect use of subsumption. For example,

ontologies might contain subsumptions incorrectly modeling

- some type of relation between two concepts, such as *Irrigation* ⊆ *Agriculture* or *Biographies* ⊆ *People*;
- part-whole relations, such as *Branch* ⊆ *Tree*; and
- role relations, such as *Garlic*, *Leek* ⊆ *Ingredient* (in fact, these are vegetables, but in some contexts they play the role of ingredient).

Inaccurate labeling led to further false mappings, such as *coal* ⊆ *industry*, where coal refers to the coal industry rather than the concept of coal itself.

Flor: Semantic enrichment of folksonomy tag spaces

Social-tagging systems such as Flickr and del.icio.us are at the forefront of the Web 2.0 phenomenon, letting users tag, organize, and share a variety of information artifacts. The lightweight structures that emerge from these tag spaces—called *folksonomies*—only weakly support content retrieval because they're agnostic to tag relationships. A search for *mammal*, for example, ignores all resources not tagged with this specific word, even if they're tagged with semantically related terms such as *lion*, *cow*, or *cat*. With Flor, our objective is to make semantic tag relationships explicit—identifying, for example, that *mammal* is more generic than *lion*—using a semantic enrichment algorithm that derives relations among implicitly interrelated tags from the Semantic Web.

We've experimentally investigated this enrichment algorithm, which builds on Scarlet. That is, given a set of implicitly related tags, our prototype identifies subsumption and disjointness relations among them and constructs a semantic structure accordingly.[12] Our experiments have furthered our understanding of Semantic Web ontologies and yielded at least two key insights. First, online ontologies have poor coverage of a variety of tag types, including those denoting novel terminology (such as Ajax and CSS), scientific terms, multilingual terms, and domain-specific jargon.

Second, online ontologies can reflect different views, and using them in combination can lead to inconsistencies in the derived structures. For example, deriving knowledge from multiple online ontologies shows that they variously consider *tomato* as a *fruit* or a *vegetable*. The first statement is valid in a biological context: a tomato is the fruit of a tomato plant. Nonetheless, many systems classify tomatoes as vegetables. Although such differing views can coexist, the fact that another ontology declares *fruit* and *vegetable* disjoint renders the derived semantic structure logically inconsistent.

Word-sense disambiguation

Jorge Gracia and his colleagues exploit large-scale semantics to tackle the WSD task.[10] They propose a novel, unsupervised, multiontology method that

- relies on dynamically identified online ontologies as sources for candidate word senses and
- employs algorithms that combine information available on both the Web and the Semantic Web to compute semantic measures among these senses and complete their disambiguation.

> Ontologies tend to be small and lightweight; the Semantic Web currently has relatively few big, dense, and large-scale ontologies.

In its early implementation, the algorithm used Swoogle to find potentially useful ontologies and then downloaded them locally for analysis. A newer version of the algorithm uses Watson to access online ontologies. Given the rich Watson API, the algorithm can access all the important information without having to download the ontologies, providing much faster functionality.

Development and use of the WSD algorithm has shown that the Semantic Web is a good source of word senses that can complement traditional resources, such as WordNet. Also, it's possible to use the extracted ontological information as a basis for relatedness computation, rather than exploit it through formal reasoning, as in ontology matching. As a result, this algorithm is less affected by formal modeling quality than Scarlet. One drawback of the WSD method, however, is that most ontologies have a weak structure; as such, they provide insufficient information to perform a satisfactory disambiguation.

So what? The Semantic Web today

Gathering and developing this range of Semantic Web applications has led us to a set of conclusions about the Semantic Web's current status.

How big? Measuring its size

The Semantic Web's size is obviously a key consideration, yet various semantic search engines estimate this seemingly simple measure differently (Sindice reports the highest value at 26 million RDF documents). Estimate variation is due to both

- a lack of agreement about what constitutes a Semantic Web document (some engines count RSS feeds, for example, and some consider each entity provided by large resources such as DBpedia as a separate document); and
- the differences in various engines' ability to identify duplicate documents.

Given this, it's difficult to give a precise estimate of the Semantic Web's size. However, we can make an educated guess that it currently contains a few million documents describing millions of entities through billions of statements. Whatever the actual size, our applications show that the Semantic Web is already big enough to make performing real-life tasks—such as aligning two large agricultural thesauri—possible. In other words, contrary to popular myths, the Semantic Web is less a long-term aspiration than a concrete reality.

How broad? Estimating its coverage

From an application perspective, the Semantic Web's topic domain coverage is an important issue. Indeed, our experience is that some domains—such as the agricultural one—offer good results, but in other domains knowledge remains insufficient. We confirmed this observation by analyzing the domains covered by the semantic

documents that Watson collected.[13] As Figure 4 shows, topics such as "computers" are well covered but others, such as "home," are almost nonexistent.

How good? Assessing its quality

The quality and richness of online knowledge will either hamper or fuel development of next-generation Semantic Web applications. All the applications we've described here depend on such quality and are each affected differently by the semantic data's quality characteristics. Indeed, Scarlet and Flor rely on exploiting formal relations and are therefore hampered by incorrect formal modeling. Such errors, however, aren't problematic for the WSD algorithm. Inversely, the WSD algorithm is hampered by weakness in online ontologies' structure—a characteristic that didn't affect Scarlet and Flor.

Analyzing a sample of the ontologies Watson collected shows that, in general, ontologies tend to be small and lightweight; the Semantic Web currently has relatively few big, dense, and large-scale ontologies.[13]

Outlook

Our experiences in developing concrete applications and analyzing Watson-retrieved documents give us concrete ideas about the Semantic Web's status, size, coverage, richness, and quality. Such experiences also inform our assessments of the Semantic Web's key issues, direction, and forthcoming developments.

Dealing with conflict and contradiction

None of our applications have a clear strategy for dealing with contradictory information derived from multiple ontologies. This is an important topic to tackle because it targets applications that exploit heterogeneous semantic resources and therefore hasn't been addressed in traditional KBS or first-generation Semantic Web applications. The notion of trust is essential here, supporting applications in selecting resources and ontologies compatible with their view and with each other.

Increasing the domain coverage

Although our work shows that the Semantic Web has a reasonable amount of available data, the sparseness phenomenon highlighted earlier indicates that we should continue the effort of encouraging and facilitating the publication of semantic data online. In particular, we should focus on providing incentive in domains where semantic technologies' added value is less apparent (that is, outside the academic and computer science worlds). Providing smart, next-generation applications that actually use this data is one way to encourage people to share their own data.

Targeting lightweight applications

As noted, most semantic documents available on the Web are small and contain lightweight knowledge. Indeed, in analyzing Watson's collection, we found that 95 percent of the online semantic documents use only a small subset of the primitives provided by ontology representation languages such as OWL—namely, the ALH(D) description logic. This doesn't mean that there's no room on the Semantic Web for applications that exploit complex logical formalisms and reasoning mechanisms. However, as our work shows, the prevalence of lightweight knowledge certainly doesn't prohibit the development of interesting new applications, as reasoning on the Semantic Web goes beyond traditional logical inferences. In addition, in our applications, intelligence is more or less as much a function of the ability to exploit large-scale knowledge sources as a consequence of sophisticated logical inferences.

Although the Semantic Web is still in its infancy, it already provides a surprising amount of useful information that various next-generation Semantic Web applications can exploit. Obviously, the infrastructure still needs further consolidation, and quality and trust are particularly severe obstacles to developing high-quality problem solvers. Nevertheless, in a short period, we've made considerable progress. Our expectation is that, as the Semantic Web infrastructure becomes more robust and more knowledge becomes available, large-scale access and exploitation of online knowledge will become the predominant paradigm for knowledge-based systems. ■

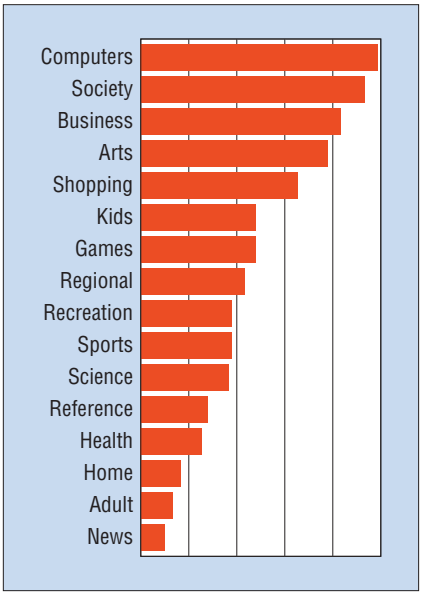

Figure 4. Relative coverage in the Semantic Web documents of the top 16 topics in the Open Directory Project's hierarchy (Directory Mozilla, or DMOZ). The *x*-axis represents a global measure of the coverage for a given topic as the sum of the measure of coverage for each semantic document Watson has collected.

Acknowledgments

The European Commission's Open Knowledge and NeOn (Life-cycle Support for Networked Ontologies) projects funded our research as part of the EC's Information Society Technologies program.

References

1. E. Motta and M. Sabou, "Next Generation Semantic Web Applications," *Proc. 1st Asian Semantic Web Conf.*, LNCS 4185, Springer, 2006, pp. 24–29.
2. I. Goldstein and S. Papert, "Artificial Intelligence, Language and the Study of Knowledge," *Cognitive Science*, vol. 1, no. 1, 1977, pp. 84–123.
3. B.C. Smith, "Reflections and Semantics in a Procedural Language," *Readings in Knowledge Representation*, Morgan Kaufmann, 1985, pp. 31–40.
4. E.A. Feigenbaum, "The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering," *Proc. 5th Int'l Joint Conf. Artificial Intelligence*, William Kaufmann, 1977, pp. 1014–1029.
5. T.R. Gruber, "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, vol. 5, no. 2, 1993, pp. 199–220.
6. E. Motta, *Reusable Components for Knowledge Modelling*, IOS Press, 1999.
7. A.T. Schreiber et al., *Engineering and Managing Knowledge: The CommonKADS Methodology*, MIT Press, 2000.
8. D. Fensel and E. Motta, "Structured Development of Problem Solving Methods," *IEEE Trans. Knowledge and Data Eng.*, vol. 13, no. 6, 2001, pp. 913–932.
9. M. Stefik, "The Next Knowledge Medium," *AI Magazine*, vol. 7, no. 1, 1986, pp. 34–46.

The Authors

Mathieu d'Aquin is a research fellow at the Open University's Knowledge Media Institute. His research interests are in tools and infrastructures for supporting the development of Semantic Web applications. d'Aquin received his PhD in computer science from the University of Nancy. Contact him at m.daquin@open.ac.uk.

Enrico Motta is a professor of knowledge technologies at the Open University's Knowledge Media Institute. His research focuses primarily on integrating semantic, Web, and language technologies to support the development of intelligent Web applications that can exploit the emerging Semantic Web's large-scale data. Motta received his PhD in artificial intelligence from the Open University and is editor in chief of the *International Journal of Human Computer Studies*. Contact him at e.motta@open.ac.uk.

Marta Sabou is a research fellow at the Open University's Knowledge Media Institute. Her research interests are in using AI and Semantic Web techniques to build applications that use large-scale semantic data. Sabou received her PhD in AI from the Free University, Amsterdam. Contact her at r.m.sabou@open.ac.uk.

Sofia Angeletou is a PhD candidate at the Open University's Knowledge Media Institute. Her research focuses on the semantic enrichment of tagging systems to enable intelligent annotation, search, and navigation. Angeletou received her diploma in computer engineering and informatics from the University of Patras. Contact her at s.angeletou@open.ac.uk.

Laurian Gridinoc is a PhD candidate at the Open University's Knowledge Media Institute. His research interests are in using novel Semantic Web interactions as background knowledge—using a mesh of ontologies to yield interesting and often unanticipated connections. Gridinoc received his master's in computational linguistics from the University of A.I. Cuza, Iasi. Contact him at l.gridinoc@open.ac.uk.

Vanessa Lopez is a research fellow at the Open University's Knowledge Media Institute, where she is also a part-time PhD student. Her research interests are in natural-language front ends to query the Semantic Web. Lopez received her MSc in computer engineering from the Technical University of Madrid. Contact her at v.lopez@open.ac.uk.

Davide Guidi is a research fellow at the Open University's Knowledge Media Institute. His research interests are in handling, reusing, and exploiting Semantic Web knowledge. Guidi received his PhD in computer science from the University of Bologna. Contact him at d.guidi@open.ac.uk.

10. J. Gracia et al., "Querying the Web: A Multiontology Disambiguation Method," *Proc. 6th Int'l Conf. Web Eng.* (ICWE 06), ACM Press, 2006, pp. 241–248.
11. M. Sabou et al., "Evaluating the Semantic Web: A Task-Based Approach," *Proc. Int'l Semantic Web Conf.*, LNCS 4825, Springer, 2007, pp. 423–437.
12. S. Angeletou et al., "Bridging the Gap between Folksonomies and the Semantic Web: An Experience Report," *Proc. Workshop Bridging the Gap between Semantic Web and Web 2.0*, Univ. of Kassel, 2007, www.kde.cs.uni-kassel.de/ws/eswc2007/proc/BridgingtheGap.pdf.
13. M. d'Aquin et al., "Characterizing Knowledge on the Semantic Web with Watson," *Proc. Int'l Workshop Evaluation of Ontologies and Ontology-Based Tools (EON)*, ISWC/ASWC, 2007, pp. 1–10, http://km.aifb.uni-karlsruhe.de/ws/eon2007/EON2007_Proceedings.pdf.