WATSON: A Gateway for the Semantic Web

Mathieu d'Aquin, Marta Sabou, Martin Dzbor, Claudio Baldassarre, Laurian Gridinoc, Sofia Angeletou, and Enrico Motta

Knowledge Media Institute (KMi)
The Open University, Milton Keynes, United Kingdom
{m.daquin, r.m.sabou, m.dzbor, c.baldassarre, l.gridinoc, s.angeletou, e.motta}@open.ac.uk

Abstract. As the Semantic Web is gaining momentum, more and more semantic data is available online. The second generation of Semantic Web applications already exploit this phenomenon by relying on this huge amount of semantic content. Looking at the requirements of these applications, we show that there is a need for an efficient access point to the Semantic Web, designed to take into account the semantic nature of the knowledge available online. However, because they rely on "classical Web" techniques, existing solutions fail to fulfill this need. In this paper, we describe the design of WATSON, a gateway for the Semantic Web, which has been guided by the requirements of Semantic Web applications and by lessons learnt from previous systems. We show how WATSON exploits the strengths of semantic technologies to provide fundamental functionalities for a more suitable access to online knowledge. We also report on using these functionalities to analyze some of the characteristics of the content of the Semantic Web.
Keywords: Semantic Web gateway, Semantic Web search engine, ontology discovery.

1 Introduction

This paper presents the design of WATSON, a tool and an infrastructure that automatically collects, analyses and indexes ontologies and semantic data available online in order to provide efficient access to this huge amount of knowledge content for Semantic Web users and applications. The Semantic Web is gaining momentum as more and more semantic data is available online. The core motivation behind WATSON is that this rapid growth will lead to the development of new applications which will need a well-suited access point, a gateway to the Semantic Web. Indeed, our analysis of a significant number of Semantic Web applications [15] has shown that a new generation of Semantic Web applications is emerging, which no more rely on a single ontology selected at design time, but rather manipulate and combine different sources of semantic content discovered at run-time. Therefore, in many scenarios, the ability to dynamically select and retrieve relevant knowledge units and semantic data is required. Moreover, the current growth of online available knowledge also leads to the possibility of studying the actual content of the Semantic Web. Indeed, a gateway to the

Semantic Web gives the opportunity to better understand to which extent semantic technologies are adopted, how they are used and in which way knowledge is actually published.

The Semantic Web is an extension of the Web and, therefore, when building a gateway for it, it seems natural to try to adapt classical Web techniques when considering semantic content. However, the requirements of the previously mentioned Semantic Web applications show that it is insufficient to rely only on a "Web view" on the Semantic Web, like it is done by Swoogle, the current state of the art semantic search engine [6]. There is a need to take into account and to exploit the particularities of Semantic Web content, adopting a truly "Semantic Web view". WATSON is based on a set of design principles inspired from this semantic oriented viewpoint.

In the remainder of this paper, we describe the design of WATSON as guided by the requirements of Semantic Web applications and by lessons learnt from previous systems. We first clarify the motivations for such a gateway, identifying the requirements of several "next generation Semantic Web applications" (Section 2). Related work is then presented, focusing on the key limitations of the state of the art search engine for the Semantic Web, Swoogle (Section 3). We then show how, by adopting a basic design principle, "going beyond the Web view", WATSON tends to overcome these limitations (Section 4). Section 5 details the concrete architecture of WATSON and shows how, by using Semantic Web techniques, we can provide advanced services to access the semantic data collected by WATSON. Finally, we report on some particularly interesting conclusions emerging from the analysis of the content of WATSON (Section 6), before pointing out conclusions and future developments (Section 7).

2 Next Generation Semantic Web Applications

We highlight two significant changes in the way recent Semantic Web applications are designed. First, they assume the existence of large scale, distributed markup that they can use, whereas earlier applications had to engineer the semantic data before using it. Another observation is that, while their ancestors integrated heterogeneous data sets under a common ontology at design time, newer applications tend to dynamically exploit the heterogeneity of semantic markup authored in terms of multiple ontologies.

The evolution of two KMi applications clearly demonstrate these new trends. The first one, AquaLog [14] is an ontology based question answering system that derives answers to questions asked in natural language by exploiting an underlying ontology. The second application, Magpie [7], is a semantic browser which assists users while they surf the Web, by highlighting instances of chosen concepts in the current Web page. To achieve this functionality, Magpie relies on an internal instantiated ontology. In both tools, the employed ontology is manually selected by the user and, while it can easily be changed, only one ontology can be exploited at a time. The new generations of these applications aim to overcome this limitation by exploiting the wealth of online semantic data.

The goal is to dynamically find and combine the relevant knowledge among online ontologies and semantic data, allowing cross-domain question answering in the case of PowerAqua [13] (the successor of AquaLog), and an extended coverage of the semantic browsing with Magpie.

This idea of exploiting the Semantic Web as a large source of background knowledge also appeared recently in several papers concerning generic ontology engineering tasks (e.g. sense disambiguation, ontology matching). For example, in his position paper at WWW'06 [1], Alani proposes a new method for ontology learning that relies on cutting and pasting ontology modules from online ontologies. Another interesting application is described in [9], where the authors propose a multi-ontology based method to disambiguate the senses of the keywords used in a search engine (e.g., in *(astronomy, start, planet)*, *star* is used in its sense of celestial body). While traditionally the authors would have relied on WordNet alone to collect possible senses for the keywords, now they can exploit all online ontologies to gather a much larger set of senses. Finally, in [20] we explored the use of online available ontologies as background knowledge for ontology matching. Our implementation identifies and combines relevant knowledge from online ontologies at run time, providing support for those scenarios where the identification of an adequate domain ontology is not possible at design time.

The success of these new applications obviously relies on the availability of a large amount of semantic data, but also requires an infrastructure for *collecting, indexing and providing intelligent access* to distributed data and ontologies on the Web. Hence, a *gateway to the Semantic Web*, an efficient entry point to online knowledge, is needed. The previously described applications use Swoogle, the state of the art Semantic Web search engine [6], but, as described further in Section 3, the design of Swoogle adopts a "Web centric" approach, leading to key limitations when considering the perspective of a *Semantic Web* gateway.

3 Related Work

The idea of providing an efficient and easy access to the Semantic Web is not new. Indeed, there have been several research efforts that have either considered the task as a whole or have concentrated on some of its sub-issues. Probably the most popular and the most advanced system is Swoogle [6], a search engine that crawls and indexes online Semantic Web documents. In this section we provide an overview of the relevant related work in this area by first concentrating on Swoogle and then describing how other systems fill in some of the gaps that were not considered by it.

Swoogle claims to adopt a *Web view on the Semantic Web* [6] and indeed, most of the techniques on which it relies are inspired by classical Web search engines. While relying on these well-studied techniques offers a range of advantages, it also constitutes a major limitation: by largely ignoring the semantic particularities of the data that it indexes, Swoogle falls short of offering the functionalities required from a truly *Semantic Web* gateway. We will now de-

scribe some of the major assumptions underlying Swoogle that stem from its Web view of the Semantic Web.

Considering only explicit relations. Swoogle only considers simple, declared relations between ontologies and other semantic documents (e.g., imports). However, unlike usual Web documents, ontologies are formalized pieces of knowledge, included in a network of implicit relations. None of the many possible relations between ontologies, such as equivalence, inclusions, versions, are considered by Swoogle. Among these relations, equivalence is the most frequent and the most important to be aware of. Indeed, many online ontologies are located at different Web addresses (URLs), because of redirection or local copies. Some ontologies may also contain exactly the same piece of knowledge (semantically), but differ in the encoding of this knowledge. Because Swoogle does not check for such syntactic or semantic duplicates, it often returns the same ontology several times with different ranking measures.

Weak notion of semantic quality. By employing a PageRank-like algorithm to order its results, Swoogle, like Web search engines, only measures the quality of its ontologies in terms of their popularity. However, in scenarios where users need ontologies for reusing or exploiting them, the quality of the ontology can be as important as its popularity (or even more). This limitation has been recognized by several researchers. Most notably, the AKTiveRank [2] algorithm employes a set of ontology structure based metrics to assess the quality and the density of the knowledge conceptualized in ontologies. Ranking ontologies by their semantic quality rather than their popularity proved to better correspond to the needs of the users. Indeed, the authors found a high level of correlation between ranks produced by AKTiveRank and those produced by users (0.952, where 1 shows identical rankings), while Swoogle ranks proved to be against those expected by the users (-0.144, where 0 means no correlation).

Weak access to semantic content. Swoogle also largely ignores the semantic nature of the indexed data when providing access mechanisms to this data: its querying facilities are limited to keyword based search (plus a couple pre-canned queries offered via Web services). However, knowledge on the Semantic Web is formalized using technologies that allow for more fine-grained and formal queries, e.g., finding all the instances of a given class or the relations occurring between two instances. Such a formal query interface is proposed by the OntoSearch2 [17] system. We think that there is a need for a wide range of access mechanisms since different applications might need different ways of querying the available data, or even might need to combine basic techniques, like keyword search, with more advanced ones.

Besides Swoogle, several other systems aim at providing an efficient access to ontologies and semantic data available online. For example, OntoKhoj [18] is an ontology portal that crawls, classifies, ranks and searches ontologies. For ranking they use the OntoRank algorithm which is in spirit similar to PageRank. Oyster [16] is different from the previously mentioned systems in the sense that it is focused on ontology sharing: users manually register ontologies and their metadata which they can then access over a peer-to-peer network of local

registries. OntoSelect [4] is an ontology library that focuses on providing natural language based access to ontologies. Finally, MultiCrawler [10] is focused on an architecture for discovering, exploring and indexing structured data on the Web.

We conclude that, while there is a significant amount of work directed towards providing access to online semantic data, existing approaches have considered only parts of the task. Our work addresses this gap, aiming at providing a complete system which is based on assumptions valid for the Semantic Web.

4 Design Principle: Beyond the Web View

In this section, we describe the major design principles on which our work relies and discuss how these are intended to overcome the limitations of existing systems discussed in the previous section.

From explicit relations to implicit semantic relations. We have discussed in Section 3 that a wide range of explicitly declared or implicit relations exist between ontologies that are largely ignored by Swoogle and other tools. Taking into account these relations is important since they partially define the semantics of the ontologies. Indeed, as explained in [10], crawling the Semantic Web needs to consider, besides classical hyperlinks relating Web pages to semantic content, a wide and extensible range of explicit links between semantic documents (e.g., `imports`, `seeAlso`) and the semantics of these relations should be exploited when collecting semantic data. Second, special attention needs to be directed towards making the implicit relations between ontologies explicit. This implies applying a wide range of analysis tasks (e.g., duplication detection) to process, compare and relate semantic documents. When made explicit, these implicit relations can be used as a basis for providing more advanced semantic clustering and navigation of semantic content.

Focusing on semantic quality. The Semantic Web is characterized by a great variety of semantic data, ranging from high quality, richly axiomatized ontologies to flat bags of factual data. While the whole range is useful, different ontologies are needed for different tasks and scenarios. Hence, information about the quality of each semantic document is crucial to provide the most relevant access to users and applications. The implication of making ontology quality a core concern of the gateway is that the collected data should undergo a *validation* process that would assess the quality of each indexed document. In addition to the properties that are usually computed for classical documents (size, encoding, etc.), validation should assess characteristics that are useful for understanding the content of Semantic Web based data, e.g., the expressivity of the employed ontology language, the level of axiomatization, etc. Another implication of this quality-driven view is that semantic documents should be indexed in accordance to their particular characteristics – e.g. whether they are rich ontologies or factual data – in order to provide relevant retrieval mechanisms for different kinds of applications. Finally, the retrieval of high quality results can be ensured by appropriate ranking mechanisms that would combine the quality measures computed during validation.

Providing rich, semantic access to data. As shown in Section 2, a variety of different applications will require access to the Semantic Web by using the gateway. These applications require different levels of formalization concerning the data they manipulate and the way to retrieve it (e.g., Magpie would send a set of terms as input, while AquaLog needs support for more fine-grained, triple level queries). This should be taken into account by providing a range of access mechanisms that combine various query specifications, ranking measures and interfaces to these mechanisms, either for humans or software agents. This involves the integration of the latest research results from the fields of ontology selection [21], ontology evaluation [3, 11] and ontology modularization [5], as shown in the next section.

5 Watson: A Gateway for the Semantic Web

The role of a gateway to the Semantic Web is to provide an efficient access point to the online ontologies and semantic data. Therefore, such a gateway plays three main roles: 1- it collects the available semantic content on the Web, 2- analyzes it to extract useful metadata and indexes, and 3- implements efficient query facilities to access the data. While these three tasks are generally at the basis of any classical Web search engine, their implementation is rather different when we deal with *semantic* content as opposed to Web pages.

5.1 The Watson Architecture

In order to address the three broad design principles set out in Section 4, Watson has been designed around three core activities, each corresponding to "a layer" of its architecture as depicted in Figure 1 and described in the remainder of this section.

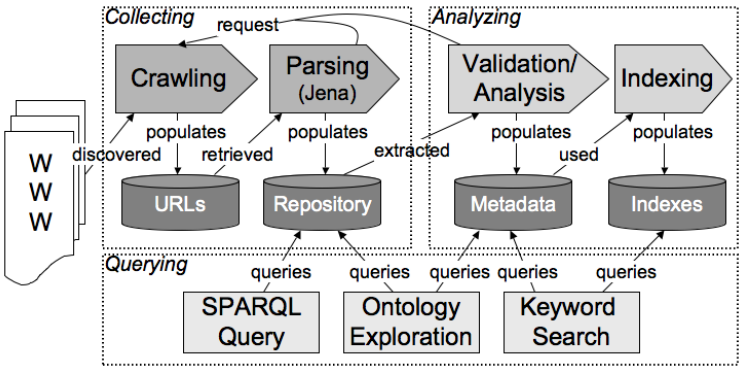

Fig. 1. A functional overview of the main components of the Watson architecture.

Ontology crawling and discovery collects the online available semantic content, in particular by exploring ontology based links

The validation and analysis layer is core to the architecture and ensures that data about the quality of the collected semantic information is computed, stored and indexed.

The query and navigation layer grants access to the indexed data through a variety of mechanisms that allow exploring its various semantic features.

Collecting Semantic Content: Crawling the Semantic Web. The goal of the crawling task in WATSON is to discover locations of semantic documents and to collect them. Classical Web crawlers can be used, but they need to be adapted to take into account the fact that we are not dealing only with Web pages, but also with semantic content. Three major questions should be considered:

1. What are the original sources of semantic content? We have to identify initial locations of documents to be explored.
2. How to inspect and explore retrieved semantic documents? We have to identify links and relations in the collected documents that would point to other documents to be crawled.
3. How to recognize a document to be considered? We need a simple procedure to eliminate documents that do not contain any semantic data.

Concerning the question of the sources of ontologies, the easiest ones to identify are existing and well-known repositories, like Swoogle or the Protégé ontology library[1]. We designed specialized crawlers for these repositories, extracting potential locations by sending queries that are intended to be covered by a large number of ontologies. For example, the keyword search facility provided by Swoogle is exploited with queries containing terms from the top most common words in the english language[2]. Another crawler heuristically explores Web pages to discover new repositories and to locate documents written in certain ontology languages (e.g. by including `filetype:owl` in a query to Google). Finally, already collected semantic documents are frequently re-crawled, to discover evolutions of known semantic content or new elements at the same location.

The retrieved ontologies are inspected in order to extract information linking to other potential locations of semantic documents. In addition to classical hyperlinks used in Web pages, there are several semantic relations between ontologies that have to be followed, either declared ones (e.g. `owl:import`, `rdfs:seeAlso`) or implicit ones (e.g. dereferencable URIs, used vocabularies).

Finally to keep only the documents that may contain semantic data or ontologies, we eliminate any document that can not be parsed by Jena[3]. In that way, only RDF based documents are considered. Further developments will also include the support of semantic data embedded in Web pages, or encoded in other ontology languages.

[1] `http://protege.stanford.edu/download/ontologies.html`
[2] `http://www.world-english.org/english500.htm`
[3] `http://jena.sourceforge.net/`

At a more technical level, the crawling layer of WATSON relies on Heritrix, the Internet Archive's Crawler[4]. Heritrix is based on a pluggable architecture; allowing us to manage different crawl profiles by setting different pipelines of custom crawling modules.

Analyzing Semantic Content: Validation, Indexing and Metadata Generation. A validation task is crucial and necessary to clean up the semantic content crawled from the Web. It requires to analyze the content of the retrieved documents in order to extract relevant information (metadata) to be used by the search functionality of WATSON. Besides trivial information, like the labels and comments of ontologies, some of these elements influence the way WATSON is designed. For instance, there are several ways to declare the URI of an ontology: as the namespace of the document, using the `xml:base` attribute, as the identifier of the ontology header, or even, if it is not declared, as the URL of the document. URIs are supposed to be unique identifiers in the scope of the Web. However, local copies of an ontology can be found at several locations (different URLs), and, as discussed further in Section 6, two ontologies that are intended to be different may declare the same URI. For these reasons, WATSON uses internal identifiers that may differ from the URI of the collected semantic documents.

Another important step in the analysis of a semantic document is to characterize it in terms of its content. WATSON extracts, exploits, and stores a large range of declared metadata or computed measures, like the employed languages/vocabularies (RDF, RDFS, RSS, FOAF, OWL, DAML+OIL, ect.), information about the contained entities (classes, properties, individuals and literals), or measures concerning the quality of the knowledge contained in the document (e.g., the expressivity of the employed language, the density of the class definitions). By combining these informations, WATSON can decide whether a particular document should be treated as a semantically rich ontology or as a more simple piece of structured data. These elements are then stored and exploited to provide advanced, quality related filtering, ranking and analysis of the collected semantic content.

In the previous paragraphs, the role the analysis task was to extract metadata concerning one particular semantic document. In addition, a core aspect in the design of WATSON concerns the exploitation of relations *between* semantic documents. Declared semantic relations (e.g., `owl:import`) are already extracted and considered at the crawling step of the process, but, due to their semantic nature, ontologies can be compared to compute their *implicit* links. Equivalence is one of the most obvious of these relations, which is nevertheless crucial to detect. Indeed, detecting duplicated knowledge ensure that we do not store redundant information and that we do not present duplicated results to the user. Two different levels of duplications are considered in WATSON. First, the files retrieved by the crawler are systematically compared together, at a purely syntactic level. Second, at a more semantic level, we compare the models of the considered ontologies, by abstracting them from their serialization, the employed language and

[4] `http://crawler.archive.org/`

syntax, or "meaningless" annotations (comments, etc.) As explained further in Section 6, this mechanism is for example able to "cluster together" documents containing the same ontology, but represented in different languages, like the ISWC ontology in DAML+OIL[5] and OWL[6].

On the same basis, several other relations are considered relying on particular notions of similarity between ontologies (inclusion, extension, overlap, etc.) Combined with other information from the crawler (e.g. date of discovery, of modification) these relations allow us to study and characterize the evolution of ontologies on the Web, through their different versions.

Querying Watson The third layer of components in the WATSON architecture takes care of the user and application front-end services. WATSON exposes its automatically gathered and validated data through a number of query interfaces, which can be characterized according to different criteria. Indeed, WATSON accepts queries ranging from classical keyword based search to formal queries in order to cater for the needs of a variety of applications. On the level of data granularity, WATSON is providing access on the level of *ontology as a whole* or on the level of *classes and other entities*. In terms of semantic content, WATSON supports querying the *declared* as well as *computed* content of ontologies and their relations. Finally, WATSON supports querying via a *human-accessible Web* interface, and partially also via a machine-accessible SparQL endpoint [19], as well as a range of specialized Web services.

The *keyword-based query* facility is similar to Swoogle's – its role is to retrieve and access semantic content including a particular "search phrase". Such a search phrase includes single or multiple keywords, keywords with wild cards and keywords arranged using logical operators. Keywords are matched against the local names, labels, comments and/or literals occurring in ontologies, using various, customizable matching functions (exact, partial, approximate, etc.) This functionality is realized via the Apache Lucene reverse indexing tool.[7] WATSON returns URIs of matched entities, which can serve as an entry point for iterative search and exploration. Indeed, a *URI-based* access allows enquiries related to entities assigned to URIs, providing a fine-grained access to the available data and to semantic relations occurring between ontology entities.

At the time of writing (December 2006), WATSON supports full keyword querying, most declared ontology relations (e.g. imports), selected computed relations (e.g. duplication and semantic similarity), and a limited sub-set of SparQL expressivity.

5.2 Advanced Services on Top of Watson Architecture

The architecture presented in the previous section can be seen as the *core* of WATSON, its kernel, providing the basic computational and storage facilities

[5] `http://annotation.semanticweb.org/iswc/iswc.daml`
[6] `http://annotation.semanticweb.org/iswc/iswc.owl`
[7] `http://lucene.apache.org`

required by a gateway for the Semantic Web. Moving up to the next level of functionality, a number of advanced techniques exist for facilitating knowledge retrieval, reuse and exploitation. In this section, we show how some of the techniques designed in KMi have either already been implemented (ontology selection and modularization) or are currently developed (ontology organization and navigation) on top of the core architecture of the gateway, with the aim of providing a range of powerful access mechanisms to the collected content.

Advanced ontology selection mechanisms. Ontology selection is the process of retrieving a list of ontologies ranked according to a set of criteria that they should satisfy (e.g., contain certain concepts). Being one of the fundamental functionalities of WATSON, selection is important both in use cases where a human user filters the results and in cases when the output is automatically integrated by applications. Our observation is that existing selection mechanisms mainly support human centered tasks and therefore fall short of supporting automatic knowledge-reuse scenarios [21]. To address the stricter requirements from this kind of scenarios, several aspects need to be considered.

First, the process of mapping the input keywords to ontology entities is usually implemented through string similarity, thus ignoring the semantics of the ontology classes given by their ontological definition (e.g., when searching for *Queen* both ontologies about bees and royal titles are returned). On the contrary, the selection mechanism described in [21] employs semantic matches built on the principle of semantically interpreting class labels based on their constituents as well as their position in the ontology.

A second issue that requires improvement relates to the way the results are ranked. We believe that ontology quality evaluation should be core to ranking [22]. However, despite the vast literature on ontology evaluation [3, 11], few of the existing selection techniques use semantic quality measures as part of their ranking scheme. The ranking mechanisms offered by WATSON rely on a combination of simple, basic quality measures that are computed in the validation phase and stored along with the ontologies (i.e., structural measures, topic relevance, etc.) These measures can be combined at run-time to offer a ranking scheme that is closest to the needs of the user/application. Note that, the pre-computed quality measures are available for querying, thus allowing other researchers to build their own ranking mechanisms without the need of computing these measures themselves (as they currently do, e.g., in AKTiveRank).

Finally, many applications require a complete coverage of the keywords they query for [21]. Our experiments indicate that the sparseness of knowledge on the Web often makes it impossible to find a single, all-covering ontology but that several ontologies can jointly cover the query terms [21]. In such cases where existing selection mechanisms would not return any result, our selection gives the smallest combination of ontologies covering the given set of terms.

Selecting only Relevant Knowledge Components. Many applications that wish to reuse online available semantic data have well defined information needs,

often requiring the retrieval of easy to integrate bits of knowledge (e.g., modules, triples) rather than entire ontologies. Unfortunately, several online ontologies are quite large (e.g., WordNet, the NCI ontology [8]) and thus, they hamper reuse activities because 1) they are often retrieved due to their wide coverage and 2) they contain a big amount of unrequested knowledge, making them difficult to exploit and to interpret. To facilitate the reuse of online knowledge, WATSON implements a range of strategies. First, simple mechanisms are used for retrieving only the description of a particular entity, the branch in the taxonomic hierarchy in which it appears or its neighborhood in the ontology graph. Second, a more advanced technique has been described in [5] for extracting the *module* of an ontology that corresponds to a set of terms. The terms used for the selection query are considered as a sub-vocabulary of the ontology, and the knowledge related to this sub-vocabulary is extracted by traversing ontology relations and properties. This technique returns small and focused knowledge components, in which the initial search terms can be easily identified and related. It is designed to be used in a fully automatic way, thus helping Semantic Web tools and applications to deal with the issues of knowledge reuse and exploitation in a scalable way.

Topic-based Ontology Organization. Understanding how ontologies relate to generic topic domains (e.g., medicine, sports) would allow us to provide a wide range of functionalities. In particular, users could perform topic based queries for ontologies and tool developers would be informed of the domains that use Semantic Web technologies. We aim to support these functionalities by incorporating mechanisms that determine the topic domain of a given ontology, thus providing an automatically generated directory of ontologies.

In our current approach, topic domains and the sets of terms that define them are established by exploring the Open Directory Project Web catalogue[8]. We use the 17 top level categories as our main topic domains. The terms that define these domains are extracted from the names of their subcategories and the keywords derived from the Web pages that they classify. We consider that an ontology belongs to a topic domain if the local names of its classes are the same as some of the defining terms for that domain. Obviously, mappings between ontologies and topic domains can be fuzzy, since ontologies can contain terms specific to several domains. In such cases, the best covered domain is considered to be representative.

Navigation and Interaction with Networked Ontologies. The functionalities reviewed in the previous paragraphs focus largely on facilitating the selection of relevant knowledge components (either complete ontologies or "modules"). In addition to these challenges, WATSON has to support human users in the task of navigating in the growing ontology space.

As mentioned earlier, WATSON currently supports querying using keywords, entity and ontology URIs. Therefore, as a number of possible configurations can

[8] `http://www.dmoz.org`

be used in querying and exploring the content of WATSON, it becomes more difficult to interpret the results. For this reason, instead of a "one view fits all purposes", we differentiate views based on different user motivations. For instance, retrieving ontologies *containing* "Turkey" suggests a different pattern than *finding classes* related to "Turkey as a country". Where in the former case it is sufficient to show a list of ontology URIs against keyword(s), in the latter case, one needs to show how the content of a particular ontology fragment and/or class neighbourhood satisfies the query.

Moreover, it is important to not only depict the ontological content (i.e. classes, individuals, etc.) but also enable the user to explore the notion of the *ontology network*. As mentioned earlier, current tools (e.g. Swoogle) do not distinguish semantic duplicates or similar content, leading to difficulties in apprehending the complex relations between ontologies, and so, in integrating heterogeneous data. Thus, an important functionality WATSON will offer is the capability to explore the set of retrieved ontologies alongside relational dimensions such as semantic duplication, overlap, extension, incompatibility, and similarly.

Finally, many user interaction tools working with ontologies support rather low-level navigation – mostly in terms of RDF graphs and *isA* hierarchies. While these clearly have their benefits, with a network of ontologies these methods rapidly lose their edge. When it is hard to visualize and make sense of a graph for a single ontology, the complexity of a network with dozens of ontologies is likely to overwhelm these visualization techniques. Hence, another challenge for WATSON will be to use such techniques as modularization, summarization or mapping to create more *content-oriented* visual metaphors.

6 The Content of Watson: Towards a Fine-Grained Characterization of the Semantic Web

Among other benefits, WATSON may take advantage of the fact that its metadata store contains not only indexing information (keyword – ontology) but a wealth of data about individual ontologies. This enables us to explore some statistical questions; e.g. *"Which ontology is most widely re-used (imported)?"* Furthermore, we can start investigating how the basic assumptions underlying the original vision of the Semantic Web (e.g., the compliance with its recommendations) are actually fulfilled in its current state. Finally, by using indications concerning for example the usage of ontology language features, some analysis can be conducted to better understand the way knowledge is represented and published on the Semantic Web. In this section, we briefly report some experiments and analysis that have been made on the current content of WATSON, pointing out particularly interesting (or surprising) elements.

One of these experiments relates to the most fundamental assumption underlying the Semantic Web: *URI are unique identifiers*. Indeed, we found that, among the semantic documents that *have not been detected as duplicates*, a small proportion (less than 1‰) declare the same URI. The analysis of these documents allowed us to identify different reasons for which URI are reused, sometimes

pointing out recommended practices (e.g. the use of `http://www.example.org` for examples[9]) and sometimes indicating common mistakes (e.g. the use of ontology language vocabularies like `http://www.w3.org/2002/07/owl` as an ontology URI). An interesting case of such a mistake is the use of the default URI proposed by the ontology editor (the one of Protégé, `http://www.owl-ontologies.com/unnamed.owl`, is among the most "popular URIs"). Finally, several cases of reusing a URI are related to practices for which there is currently no clear recommendation, like in different versions of an ontology, or in parts of the same ontologies distributed in several documents.

A simple overview of results of the mechanisms employed by WATSON for detecting duplicate ontologies show that there is a very high level of redundancy on the Semantic Web (almost 20% of the collected semantic content is redundant). For example, Swoogle claims to provide 119 results for the query "student university researcher"[10], but even in the first page, several ontologies would be clustered together by WATSON. For instance:

- `http://139.91.183.30:9090/RDF/VRP/Examples/ka2.rdf` and `http://athena.ics.forth.gr:9090/RDF/VRP/Examples/ka2.rdf` are obviously the same file at different locations
- `http://annotation.semanticweb.org/iswc/iswc.owl` and `http://annotation.semanticweb.org/iswc/iswc.daml` contain the same knowledge but in different languages

Measuring how Semantic Web languages are used can provide valuable feedback for the designers of these technologies. In particular, as already shown by [23] on a set of 1300 ontologies, it can be seen that an important proportion of the OWL ontologies collected by WATSON fall into the OWL Full species, even if the expressivity of the language is largely under-exploited (they rarely use more than an $\mathcal{AL}(D)$ description logic). Additional analysis (in particular using data-mining techniques) can be used to extract more fine grained indications from our metadata store. It is for example interesting to know that *cardinality restrictions* are rarely used compared to other restrictions in OWL, but are very often used alone, without the other OWL class constructors.

The huge amount of semantic data and metadata collected by WATSON gives the opportunity to study many other aspects of the Semantic Web, like the use of multilinguality features or the presence of inconsistencies and contradictions in interconnected ontologies. The experiments reported here provide initial examples of how WATSON can be used as a research platform, helping researchers and developers to better understand the Semantic Web and its evolution.

7 Summary

In this paper, we have presented the design of WATSON, a *gateway* which provides efficient access to the content of the Semantic Web by addressing the requirements of emerging Semantic Web applications. At the time of writing this

[9] `http://www.rfc-editor.org/rfc/rfc2606.txt`

[10] results obtained the 14th December 2006.

paper, the core architecture of WATSON (Section 5.1) has been implemented as a prototype. Several advanced services (Section 5.2) have already been developed on top of this architecture (ontology selection, modularization) or are currently under development (ontology organization, navigation in networked ontologies). An important effort is currently being devoted to make this prototype version evolve to a scalable, robust and publicly available implementation, which we plan to release by early spring.

By exploiting the semantic nature of ontologies, WATSON has already demonstrated its ability to go beyond the functionalities provided by related systems. Indeed, detecting duplicated and related knowledge, analyzing and exploiting the qualities of semantic data, or providing multiple and adapted querying methods are major strengths of WATSON compared to existing search engines like Swoogle. Because of these *design principles*, WATSON can be used to select and retrieve more relevant, adequate and easy to exploit knowledge, thus truly playing the role of a gateway to the Semantic Web.

Finally, another direction in which WATSON can exploit the Semantic Web infrastructure concerns the interoperation with other systems. Several applications, in particular the ones developed in KMi (Magpie [7], PowerAqua [13]) are evolving to use WATSON for selecting, retrieving and exploring semantic data and ontologies. In addition, we plan to exploit the interoperability allowed by Semantic Web technologies for integrating functionalities provided by other systems in WATSON. In particular, we are currently working on the interaction with Oyster [16], for making ontologies collected by WATSON available in a peer-to-peer setting. Another valuable extension would be the ability for the user to review and rate available semantic content. This functionality is planned to be implemented in a near future by relying on the `revyu.com` website [12].

Acknowledgements. This work was funded by the Open Knowledge and NeOn projects sponsored under EC grant numbers IST-FF6-027253 and IST-FF6-027595.

References

1. H. Alani. Position Paper: Ontology Construction from Online Ontologies. In *Proc. of the 15th International World Wide Web Conference*, 2006.
2. H. Alani, C. Brewster, and N. Shadbolt. Ranking Ontologies with AKTiveRank. In *Proc. of the International Semantic Web Conference, ISWC*, 2006.
3. J. Brank, M. Grobelnik, and D. Mladenic. A survey of ontology evaluation techniques. In *In Proc. of Data Mining and Data Warehouses (SiKDD)*, 2005.
4. P. Buitelaar, T. Eigner, and T. Declerck. OntoSelect: A Dynamic Ontology Library with Support for Ontology Selection. In *Proc. of the Demo Session at the International Semantic Web Conference, ISWC*. 2004.
5. M. d'Aquin, M. Sabou, and E. Motta. Modularization: a Key for the Dynamic Selection of Relevant Knowledge Components. In *Proc. of the ISWC 2006 Workshop on Modular Ontologies*, 2006.

6. L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari. Finding and Ranking Knowledge on the Semantic Web. In Y. Gil, E. Motta, V.R. Benjamins, and M.A. Musen, editors, *Proc. of the 4th International Semantic Web Conference*, volume 3729 of *LNCS*, pages 156 – 170, Galway, Ireland, November 6-10 2005. Springer-Verlag GmbH.
7. M. Dzbor, J. Domingue, and E. Motta. Magpie - towards a semantic web browser. In *Proc. of the Second International Semantic Web Conference*, 2003.
8. J. Golbeck, G. Fragoso, F. Hartel, J. Hendler, B. Parsia, and J. Oberthaler. The national cancer institute's thesaurus and ontology. *Journal of Web Semantics*, 1(1), 2003.
9. J. Gracia, R. Trillo, M. Espinoza, and E. Mena. Querying the Web: A Multiontology Disambiguation Method. In *Proc. of the Sixth International Conference on Web Engineering (ICWE'06), Palo Alto, California (USA)*. ACM, July 2006.
10. A. Harth, J. Umbrich, and S. Decker. MultiCrawler: A Pipelined Architecture for Crawling and Indexing Semantic Web Data. In *Proc. of the International Semantic Web Conference, ISWC*, 2006.
11. J. Hartmann, Y. Sure, A. Giboin, D. Maynard, M. C. Suarez-Figueroa, and R. Cuel. Methods for ontology evaluation. Knowledge Web Deliverable D1.2.3, 2005.
12. T. Heath and E. Motta. Reviews and Ratings on the Semantic Web. In *Poster Track of the International Semantic Web Conference, ISWC*, 2006.
13. V. Lopez, E. Motta, and V. Uren. PowerAqua: Fishing the Semantic Web. In *Proc. of the European Semantic Web Conference, ESWC*, 2006.
14. V. Lopez, M. Pasin, and E. Motta. AquaLog: An Ontology-portable Question Answering System for the Semantic Web. In *Proc. of the European Semantic Web Conference, ESWC*, 2005.
15. E. Motta and M. Sabou. Next Generation Semantic Web Applications. In *Proc. of the 1st Asian Semantic Web Conference (ASWC)*, 2006.
16. R. Palma, P. Haase, and A. Gómez-Pérez. Oyster: sharing and re-using ontologies in a peer-to-peer community. In *Proc. of the 15th International Conference on World Wide Web, WWW*, 2006.
17. J. Z. Pan, E. Thomas, and D. Sleeman. ONTOSEARCH2: Searching and Querying Web Ontologies. In *Proc. of the IADIS International Conference WWW/Internet*, 2006.
18. C. Patel, K. Supekar, Y. Lee, and E. K. Park. OntoKhoj: A Semantic Web Portal for Ontology Searching, Ranking and Classification. In *Proc. of the Workshop On Web Information And Data Management*. ACM, 2003.
19. Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. W3C Working Draft, World Wide Web Consortium (W3C), 2006. Accessed from http://www.w3.org/TR/rdf-sparql-query.
20. M. Sabou, M. d'Aquin, and E. Motta. Using the Semantic Web as Background Knowledge for Ontology Mapping. In *Proc. of the ISWC Ontology Matching Workshop collocated*, 2006.
21. M. Sabou, V. Lopez, and E. Motta. Ontology Selection on the Real Semantic Web: How to Cover the Queens Birthday Dinner? In *Proc. of the European Knowledge Acquisition Workshop (EKAW)*, Podebrady, Czech Republic, 2nd-6th October 2006.
22. M. Sabou, V. Lopez, E. Motta, and V. Uren. Ontology Selection: Ontology Evaluation on the Real Semantic Web. In *Proc. of the WWW Evaluation of Ontologies on the Web Workshop*, 2006.
23. T. D. Wang, B. Parsia, and J. Hendler. A Survey of the Web Ontology Landscape. In *Proc. of the International Semantic Web Conference, ISWC*, 2006.