Magpie: Experiences in supporting Semantic Web browsing

Martin Dzbor [*], Enrico Motta, John Domingue

Knowledge Media Institute, The Open University, Milton Keynes, UK

Abstract

Magpie has been one of the first truly effective approaches to bringing semantics into the web browsing experience. The key innovation brought by Magpie was the replacement of a manual annotation process by an automatically associated ontology-based semantic layer over web resources, which ensured added value at no cost for the user. Magpie also differs from older open hypermedia systems: its associations between entities in a web page and semantic concepts from an ontology enable link typing and subsequent interpretation of the resource. The semantic layer in Magpie also facilitates locating semantic services and making them available to the user, so that they can be manually activated by a user or opportunistically triggered when appropriate patterns are encountered during browsing. In this paper we track the evolution of Magpie as a technology for developing open and flexible Semantic Web applications. Magpie emerged from our research into user-accessible Semantic Web, and we use this viewpoint to assess the role of tools like Magpie in making semantic content useful for ordinary users. We see such tools as crucial in bootstrapping the Semantic Web through the automation of the knowledge generation process.

Keywords: Semantic Web; Navigation; User interaction; Semantic browsing; Annotation; Services; Application development framework

1. Introduction

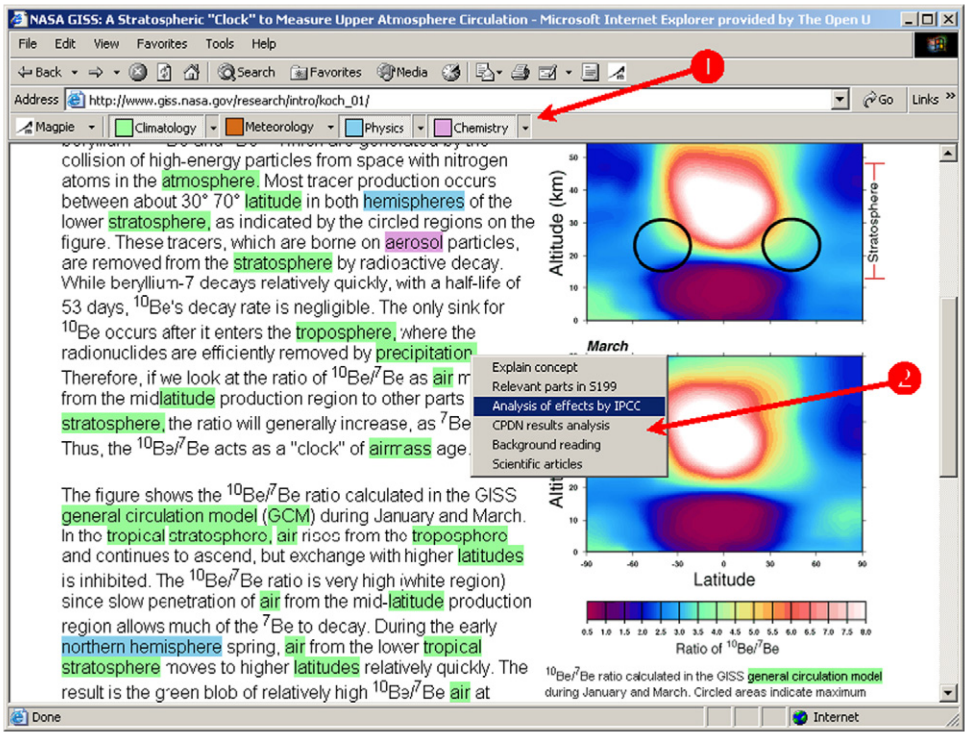
The World Wide Web is acknowledged as one of the greatest inventions of the 20th century. Its success has been built on its scalable architecture and the simplicity of its mechanisms for locating, browsing and publishing information. Despite this great success, the first generation Web suffers from a number of limitations. In particular, while the Web metaphor suits humans well, the mark-up of content using HTML, which is essentially a language for rendering information on screen, provides little support for the automatic analysis and aggregation of information and services. To address this limitation the vision of a *Semantic Web* [3] has been proposed, in which web resources are annotated with semantic mark-up, using knowledge representation languages, such as RDF(S) [5] or OWL [50]. These can be used to express formal statements about web resources, external entities, and their relationships.

Compared to HTML, these semantic mark-up languages exhibit a fair degree of complexity. It is a non-trivial exercise for an average web content provider, who is not an expert in knowledge representation, to semantically annotate a document. In addition, in order to enable agent interoperability, annotations need to be based on some particular *ontology* [27], which can cause an extra overhead for users. Hence, as in the case of the traditional *knowledge acquisition bottleneck* in knowledge engineering [30], there is a tension between the dependence of the Semantic Web on semantic annotations (done on a large scale), and the cost and complexity of providing these semantic annotations.

In addition to the knowledge acquisition bottleneck, there is another challenge worth considering—to foster the uptake of Semantic Web technologies; namely, how can semantics be brought into standard web activities and tasks, such as browsing, to provide added value to ordinary users? And how can we achieve this goal in a scenario where most web pages are not semantically annotated?

In order to address these issues (thus being able to build demonstrations of Semantic Web-like capabilities in the absence of the key ingredient: semantic mark-up) we developed an initial version of *Magpie* [18–20]—a suite of tools supporting a "low-cost" approach to annotating documents for the Semantic Web with the aim to support browsing. The key feature of Magpie was its capability to support the annotation and subse-

[*] Corresponding author.
E-mail addresses: M.Dzbor@open.ac.uk (M. Dzbor), E.Motta@open.ac.uk (E. Motta), J.B.Domingue@open.ac.uk (J. Domingue).


NASA GISS: A Stratospheric "Clock" to Measure Upper Atmosphere Circulation - Microsoft Internet Explorer provided by The Open U
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.giss.nasa.gov/research/intro/koch_01/
Go Links
Magpie Climatology Meteorology Physics Chemistry
1
2
Explain concept
Relevant parts in S199
Analysis of effects by IPCC
CPDN results analysis
Background reading
Scientific articles
Altitude (km)
Stratosphere
March
Latitude
Ratio of ^{10}Be/^{7}Be
Done
Internet

collision of high-energy particles from space with nitrogen atoms in the atmosphere. Most tracer production occurs between about 30° 70° latitude in both hemispheres of the lower stratosphere, as indicated by the circled regions on the figure. These tracers, which are borne on aerosol particles, are removed from the stratosphere by radioactive decay. While beryllium-7 decays relatively quickly, with a half-life of 53 days, ^{10}Be's decay rate is negligible. The only sink for ^{10}Be occurs after it enters the troposphere, where the radionuclides are efficiently removed by precipitation. Therefore, if we look at the ratio of ^{10}Be/^{7}Be as air m from the midlatitude production region to other parts stratosphere, the ratio will generally increase, as ^{7}Be Thus, the ^{10}Be/^{7}Be acts as a "clock" of airmass age.

The figure shows the ^{10}Be/^{7}Be ratio calculated in the GISS general circulation model (GCM) during January and March. In the tropical stratosphere, air rises from the troposphere and continues to ascend, but exchange with higher latitudes is inhibited. The ^{10}Be/^{7}Be ratio is very high (white region) since slow penetration of air from the mid-latitude production region allows much of the ^{7}Be to decay. During the early northern hemisphere spring, air from the lower tropical stratosphere moves to higher latitudes relatively quickly. The result is the green blob of relatively high ^{10}Be/^{7}Be air at

^{10}Be/^{7}Be ratio calculated in the GISS general circulation model during January and March. Circled areas indicate maximum

Fig. 1. A web page related to climate science with Magpie plug-in highlighting concepts relevant from the perspective of climatology course for a particular student. The menu shown in the center is associated with the concept of "*precipitation*".

quent interpretation of web documents with no a priori mark-up. To this purpose, Magpie used a gazetteer approach to add an ontology-derived semantic layer to the original web page—a typical screenshot is shown in Fig. 1. In addition, in order to go beyond simple mark-up and link semantics to concrete services, Magpie also supported the association of *Semantic Web services* [1,47] to the concepts highlighted in the web page. These services enabled the user to re-use the familiar web browsing and *link clicking metaphors* to navigate from a highlighted concept to other concepts, in response to semantic inference rather than syntactic connections.

The emphasis in Magpie on a user-centric, low cost approach can be observed on several levels. For example, Magpie extends a *standard web browser* to avoid the need for users to learn new tools. Moreover, the default Magpie annotation engine provides *fast*, *real-time mark-up*, to avoid any time-related overhead for its users. In contrast with other similar tools developed in the early days of the Semantic Web, such as COHSE [9] and KIM [42], Magpie offers greater flexibility, (i) by allowing users to select which ontology they want to use to provide a semantic context to their browsing (rather than being given a single ontology, as in KIM) and (ii) by moving away from the use of semantic mark-up simply as a way to support dynamic hypermedia, as in COHSE. As we suggested above, semantic mark-up in Magpie is dynamically linked to *a set of semantic services*—i.e. viewpoint-dependent actions or inference methods, which can be invoked by the user through a right-click, contextual menu, which is customized for each ontological viewpoint. This generic capability makes it possible to build Magpie-based applications, which link dynamically the information found on the Web with relevant background knowledge from ontologies in a flexible, generic way.

Although, as emphasized above, Magpie was one of the pioneering applications in the emerging area of Semantic Web tools for the end users, a number of other tools have since emerged, which support users in annotating documents [33]; in browsing semantically marked up spaces [7,36,43]; in navigating and managing semantic data or semantic services repositories [1,47]; or in integrating web and Semantic Web resources [21,32]. If we were to highlight one defining feature, then all these tools would be characterized by their emphasis on *applying* knowledge models in common, everyday situations by ordinary users—as opposed to *engineering* such models.

Given this context, the aim of this article is to aggregate a range of partial outcomes and achievements from the work on the Magpie framework since 2003, and subsequently to reflect on how our views on what constitutes a Semantic Web browsing application have developed over the past years. We then use this analysis to draw and discuss some conclusions about research directions in this area. In addition to already published accounts of Magpie, this paper aims to provide a novel view of the system and its evolution embedded in a range of practical issues. It also describes previously unpublished experiences with a large-scale application, which has been deployed in a real-life e-learning context. What this article adds to our earlier works [18–20] are the generalizations of specific technological and design decisions (in particular, Sections 3, 4 and 8 present analyses not published elsewhere). Sections 2 and 5 draw on the previously published details on Magpie and act as a glue to connect these new aspects.

2. Magpie-based Semantic Web applications

In this section we introduce the basic functionalities of Magpie and the benefits that a tool of this kind can bring to concrete users in a concrete real-world scenario. In the scenario introduced first in [20] we discuss the use of Magpie as a dynamic educational resource supporting undergraduate students at The Open University (UK). The core role and objective of this Magpie-based application was to enable students to interpret third-party material on the Web (e.g. scientific reports) related to the course theme and thus to explore the broader space of the course-related knowledge.

2.1. *Supporting climate science students with Magpie*

The Open University students enrolling in a level-one climatology course receive the usual printed and multimedia educational material. This is enriched by a computational support, which includes climate modeling software, a web-based discussion forum, a web-based electronic newsletter and other student support facilities. In addition to these internal resources, the students are expected to use web resources that include scientific analyses and reports from climate researchers, as well as news stories related to the subject. Magpie assists these students by facilitating a course-specific perspective on such texts. It enables students to relate the third-party documents to the relevant course concepts, materials, activities, and generally, to knowledge they are expected to acquire from studying the course.

Fig. 1 shows a student's web browser with a typical page describing a climate-related issue—an original contribution by NASA's Goddard Institute for Space Studies (http://www.giss.nasa.gov/research/intro/koch_01/). This is a relevant but fairly complex text for an undergraduate student, so the student interacts with it using the Magpie plug-in. As the student access the web page, this is automatically annotated with course-specific ontological concepts. These appear in response to a student selecting one or more of the ontology-specific toolbar buttons (see marker ❶). In this particular application the student can highlight concepts in four broad scientific areas: *Climatology*, *Meteorology*, *Physics*, and *Chemistry*. Highlighted concepts become "hotspots" that allow the user to request a menu with a set of actions for a relevant item. In Fig. 1, the right-click with a mouse on the term "*precipitation*" reveals a menu of semantic services (marker ❷). The menu choices depend on the ontological classification of a particular concept in the selected ontology and on what services are available for the concepts in a given ontology.

The Magpie toolbar – in this particular case the four buttons marked by ❶ in Fig. 1 – corresponds to what the Magpie framework labels as "top-level classes". These do not have to be top-level in terms of their ontological abstraction; they are top-level in terms of a specific role they have in the user's interaction with this particular ontology for climate science. The purpose of the climate science ontology designers was to emphasize the point that climatology is actually a specialization of physics, chemistry and other sciences. As climatologists often re-use the conceptual apparatus of physicists and chemists, so this particular educational ontology reflects the objective of the course team to exhibit the linkages between climatology, on one hand, and the related sciences, on the other.

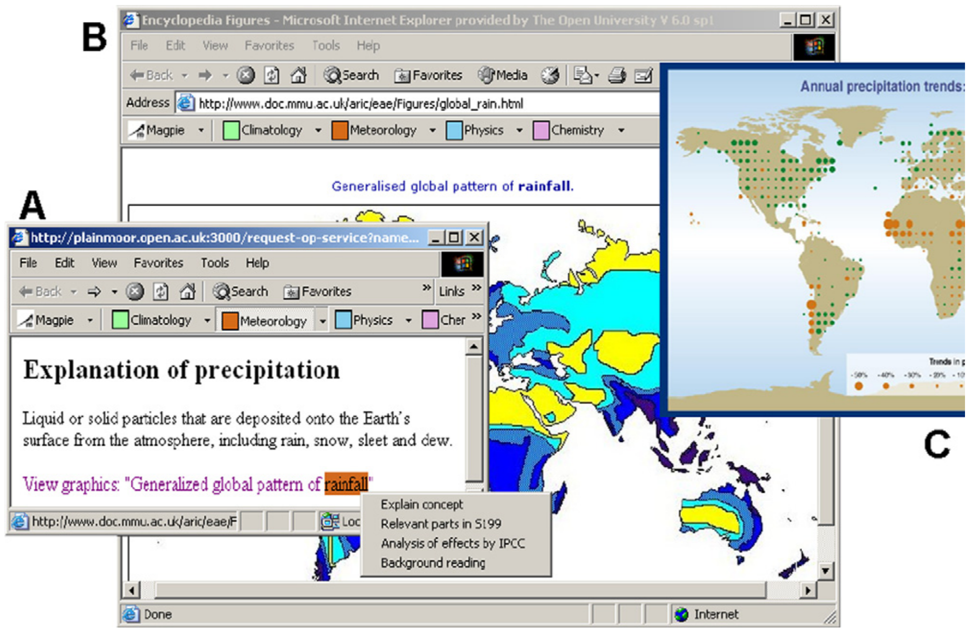
2.2. *Positives of the Magpie approach*

Since the Magpie's glossary takes in consideration the position of climate science in relationship with a specific *sub-set of* scientific disciplines, this can be seen as featuring one particular ontological viewpoint on the world related to climate. It is possible that other authors would structure the same domain differently—in their specific viewpoint, a different set of "top-level" categories may become available to the users to interact with in the form of Magpie buttons. For example, an alternative viewpoint may feature climate catastrophes, and climate-related regulations.

Without delving too deep into the area of ontology development, it suffices to say that the Magpie framework is open and transparent to these types of ontological commitments. For the user of a Magpie-enabled web browser, the choice of ontological perspective alters the appearance of the graphical user interface—the toolbar buttons. Users can at any time switch to a different ontology if it is available to them—for instance, if they want to investigate the same content/document in a slightly different context. This is one of the important design features, as it allows one to circumscribe the interpretative scope and the viewpoint on a particular problem.

The service-oriented Magpie framework supports the composition of semantic menus from the services, web services and Semantic Web services available for a particular ontology. These services can be in principle implemented by different knowledge providers. For instance, the service "*Relevant parts in S199*" shown in the semantic menu in Fig. 1 is an index to the course material provided internally by The Open University. On the contrary, the "*Background reading*" service is provided by a different university that uses a proprietary encyclopedia to provide a contextually related reading on a range of topics. Yet another type of service is "*Explain concept*". This is an aggregating service using ontology-based reasoning to combine chunks of textual and visual knowledge describing a particular concept. The aggregation is based on having a set of simpler services retrieving semantically annotated knowledge chunks from several sources and determining their semantic closeness. Naturally, the degree of sophistication of the services is independent of the Magpie architecture, which considers all services as black boxes.

As an illustration, consider the simple "*Explain concept*" service—this generates a textual explanation from an online glossary, and attaches the link to a related image, if this exists in its repository of annotated materials (e.g. Fig. 2B). The answer as shown in Fig. 2A does not explicitly exist in the course books, and indeed it is an interpretative viewpoint of the selected ontology. It facilitates an expert's view—as if a tutor was associating different materials together. Because the answer to a semantic query may be a web resource in its own right, it can be further browsed or annotated semantically. Here Magpie merges


B
Encyclopedia Figures - Microsoft Internet Explorer provided by The Open University V 6.0 sp1
http://www.doc.mmu.ac.uk/aric/eae/Figures/global_rain.html
Magpie
Climatology
Meteorology
Physics
Chemistry
Generalised global pattern of rainfall.
A
http://plainmoor.open.ac.uk:3000/request-op-service?name...
Explain concept
Relevant parts in S199
Analysis of effects by IPCC
Background reading
C
Annual precipitation trends:
Internet

Explanation of precipitation

Liquid or solid particles that are deposited onto the Earth's surface from the atmosphere, including rain, snow, sleet and dew.

View graphics: "Generalized global pattern of rainfall"

Fig. 2. An illustration of the "*Explain concept*" semantic query invoked for the "*precipitation*" concept by the semantic menu action depicted in Fig. 1. Window A shows a brief explanation acquired from the course's online glossary and a link to the associated image originating at a third-party site. The actual image chosen due to semantic proximity of its annotation to the concept is in window B. Window C shows a sample analysis relevant to the same concept by Intergovernmental Panel on Climate Change (response to another service of the menu).

the independent mechanisms for recognizing semantic relevance and browsing the resulting web resources in a manner similar to, e.g. COHSE [9] or VIeWs [7].

Unlike the above tools, Magpie also supports *trigger services*, which are based on the "subscribe&acquire" rather than the "click&go" user-system interaction paradigm. However, these were not used in the climate science application, so we leave them aside at this moment, and will return to them later, in Section 4.3.

Let us thus summarize the benefits offered by Magpie for the end users that we have considered so far. First, a domain-specific ontology supports students in making sense of information about climate science, independently of where this information resides on the Web. The application is built by selecting or constructing the appropriate ontology and by defining the appropriate services.

Because no a priori mark-up is needed, every time a new piece of information about climate appears on the Web (e.g. a news story), students can access and make sense of it with the help of the services provided by Magpie. Because the framework is service-based, every time new services related to climate are developed or updated, they can also be easily brought into the application.

2.3. *Shortcomings of the Magpie approach*

When applied to a very specific educational scenario, Magpie was considered as an innovative tool, delivering a new perspective on the use of semantics to interpret generic information available on the Web. However, as the application was rolled out to the users, we became increasingly aware of its limitations. Although the "single ontology—single interpretative perspective" paradigm used by Magpie reduces the size of the problem space, this reduction is not always helpful. This paradigm might focus the student's attention (as intended), but it also unduly restricts the breadth of the acquired knowledge (this was clearly not intended).

During a study session a student may come across a range of *similar* but semantically not entirely identical study materials. This means that at each page, the student would benefit from minor tuning of the used ontology, glossary and/or service menus. These tunings would reflect slight shifts within a broader problem space, which is a fairly common tactic we use in our everyday thinking to deal with such open situations. Thus, there was a gap between the inherent notions of a single, formal, semantically sound but "closed" ontology that guaranteed a certain level of precision within the domain, and the desire to open up the interaction by supporting multiple services, as well as multiple ontologies.

In the next section we generalize this evolution from a single ontology-based tool to increasingly open applications with relation to the Semantic Web in general. We highlight the common roots of three rather different views on what a Semantic Web application might be comprised of. Rather than presenting these views as contrasting with each other, our experiences with Magpie lead us to believe that there is a certain timeline between these views—more advanced and "open" models and approaches can be better understood via shortcomings and failures of simpler, "closed" models. Furthermore, in Section 8 we re-visit our experiences and offer our vision of the way research on semantics-enhanced browsing could evolve.

3. Views on the web of semantics

As illustrated in Section 2, Magpie assigns a semantic layer based on a user-selected ontology to an arbitrary web page. There are many conceptual ways to characterize the underlying Magpie technology. In this section we take a discursive approach to characterizing the Magpie-based applications for browsing. We take three distinct views and briefly give a rationale that helped to drive the evolution of Magpie from a single-purpose plug-in to a generic framework for developing Semantic Web applications.

3.1. Three views on Magpie-based semantic applications

One view, introduced in the first publication [17], is to see Magpie as a tool supporting the *interpretation of web pages*. The automatic recognition of entities in web pages is emphasized, and linking entities to semantic concepts is seen as a way to bring an interpretative context to bear, which, in turn, can help users to make sense of the information presented in a web page. In this case, a Semantic Web browsing tool like Magpie occupies a similar segment as, e.g. SCORE [47]. For instance, in the context of a climate science course, the purpose of a semantically enriched web browser was indeed to adopt the viewpoint of an expert in the field.

Another way to look at Magpie is as a more generic *Semantic Web browser*, and this can be seen as the second stage in the evolution of the Magpie technology [20]. If we take this view, then a browsing tool like Magpie provides an efficient way to integrate semantic and "standard" (i.e. non-semantic) approaches to browsing the Web. This is achieved through an automatic association of semantic annotations with web pages and through the provision of a simple yet effective user interface. The familiar, web browser-based user interface allows the user to navigate the resources using both hypertext and semantic links. The concept of browsing using semantic relationships, conceptually proposed in the hypertext research community and piloted by COHSE [9], essentially extends a familiar metaphor from the standard Web. It allows the user to move between the chunks of information based on semantic relations or semantic proximity, rather than following physical (mostly author- or editor-defined) hyperlinks.

One aspect that influences the capability of a semantic browser to navigate between semantically annotated entities is related to *recognizing concepts and entities in the free text* of standard, non-annotated web documents. The Magpie application in the scenario described in Section 2 uses a gazetteer approach. Instead of a static list of items, Magpie derives the lexicon (gazetteer) from an ontology, and uses this dynamic, user-selected structure as an input for recognizing and marking up entities that are relevant to a particular viewpoint.

The derivation of lexicons from ontologies made the gazetteer approach more flexible and usable in different contexts. Having said so, the application of Magpie in the climatology course merely recalled existing knowledge of the domain that had already been conceptualized in a specific ontology. In terms of offering semantic services, this Magpie application epitomizes a traditional approach to building knowledge-based systems (KBS)—knowledge is acquired from the experts at the beginning, it is represented using domain ontology, and it is not modified during the life of the application without another explicit knowledge acquisition process.

In reality, knowledge is subject to constant evolution—e.g. new concepts may emerge in the domain, or the original acquisition may have missed some domain concepts and terms. If a semantic browser is based on a single, handcrafted ontology, it may soon become obsolete. Magpie's capability to (re-)load its lexicons only partly resolved the problem. A greater issue remained with adding new entities and new knowledge; thus, adapting the lexicons to the browsing context.

Being able to recognize not only known entities, concepts and terms in a web page, but also potentially relevant extensions and additions, is an important facet of Semantic Web browsers, and the focus of current research in information extraction (IE) [11,12,15,22]. This capability is closely related to our decision to base Magpie's semantic browsing on the notion of semantic services rather than directly on the semantic relations or properties.

We believe that the second functionality a semantic browser should exhibit is the *provision of services that facilitate navigation* alongside semantic links. Here is the main difference between open hypermedia and the Magpie as a Semantic Web tool. While open hypermedia were largely document- and information-centric, Magpie services are focusing on tasks and activities. Thus, invoking a service in Magpie serves a dual purpose. On the one hand, it covers the open hypermedia functionality of dynamically linking to a new, potentially dynamically created web page. On the other hand, however, a service often executes a complex task, and provides the results from multiple sub-tasks and multiple information sources aggregated into a coherent set of knowledge-level statements. We will return to the capability of executing and combining tasks in Sections 4.1–4.3, where we discuss the benefits an open, service-based architecture offers to automating the knowledge acquisition process.

The third viewpoint, and the third evolutionary stage, we can use to characterize Magpie, is as a *framework for developing Semantic Web applications* [21]. According to this view, the Magpie suite of tools can be seen as a "shell" for building Semantic Web applications, which provides generic mechanisms to bring together ontologies, web resources and (semantic) web services. For instance, the climate science example from Section 2 can be viewed as a Semantic Web application; it is characterized by a range of existing web resources/documents, a formal domain ontology, and a number of ontology-based services, which are made available to students opportunistically, when the "right web page" (or better, the right concept) is encountered.

The key feature of Magpie in the role of a framework for Semantic Web applications is that it allows developers to focus on the *semantic* functionalities, i.e. specifying and populating the ontology, and defining the services, with no need to identify, let alone annotate web resources. Moreover, much of the user interaction management is taken care by the core Magpie framework, which further simplifies the design and the deployment of a semantic application. This is also beneficial for the developers and researchers, because the majority of them are more active in such areas as formal knowledge representation, knowledge acquisition and reasoning, or web service development, rather than in the user–ontology interaction. Other frameworks occupying a similar segment as Magpie in its role of an application development framework include, for instance, Semantic Content Organization and Retrieval Engine (SCORE) [47] from Semagix or the UIMA project [25] from IBM.

3.2. *What constitutes a Semantic Web application*

Whatever viewpoint we take on Magpie, the key aspect of this research, and of the Magpie evolution, is that it neatly captures the challenges emerging from the progress of Semantic Web research over the last few years. In addition to challenges such as knowledge representation, efficient reasoning, information extraction, annotation or expert-driven knowledge acquisition, we can align our experiences with Magpie and the new challenges—for example, the capability to handle multiple ontologies and to reconcile conceptual differences among them by mapping, aligning or merging, or the capability to operationalize the available conceptual knowledge by means of procedures, rules or services. We believe that these new trends and challenges are critical for future evolution, and indeed success, of the Semantic Web.

In this section we would like to present a set of features or characteristics making a Semantic Web application—with a particular emphasis on Semantic Web applications supporting browsing and integration with the resources on the Web. This is not the first attempt on such a characteristics. For example, Quan and Karger [44] suggest that the primary functionality an application for the Semantic Web should satisfy is "to separate the content – the proper purview of the publisher serving the information – from presentation – an issue in which the end user or their local application should have substantial say".

Based on our experiences with Magpie and the three views on this technology discussed earlier, we suggest a few features that may, in our opinion, constitute a Semantic Web application (i.e. as opposed to merely web applications). A Semantic Web application aimed at and supporting ordinary users ought to satisfy a number of criteria, listed and briefly justified below:

1 *It has to work with ontologies to ground the functionality in some domain*.

This criterion draws attention to two separate points. First, merely tagged or annotated concepts might be very useful, but there is no defined and shared semantic meaning in such annotations, unless the users explicitly subscribe to a specific formal model—ontology.

Second, the criterion uses plural "ontologies" to emphasize that the user of an application should have a choice in deciding not only how the mark-up is done and presented, but also how it is conceptually grounded. In other words, the same concept may acquire different roles, if a different ontology is chosen to interpret it. Many examples of this can be given, to mention one as an illustration: "temperature" may have a connotation, e.g. of a disease symptom of influenza in a medical ontology, but it in a climate ontology could be a consequence of a particular climate phenomenon such as greenhouse effect.

The user of a Semantic Web application shall be able to choose the conceptual frame (ontology) for interpreting and grounding various user interactions with annotations grounded in a particular frame (ontology). This criterion can be phrased also in terms of parametrizing a Semantic Web application and customizing some of its generic functionalities with multiple ontologies. In other words, ontologies cease to be hard-coded into the tools, and may even be integrated on the fly from several independent ontological modules. There are already emerging tools that offer support for working with many ontologies; e.g. the Swoogle [16] or Watson [14] ontology gateways (repositories).

2 *It has to be able to use different views on the* (*populated*) *ontologies*.

While a Semantic Web application may not necessarily populate a chosen ontology, it shall be capable of handling instances in ontologies and of customizing the way these instances are presented to the user. What this criterion emphasizes is the need to go beyond low-level representations of ontological instances. For example, instead of merely showing how the concept of "Visiting Professor" is defined formally, it might be often more helpful to enable user-oriented descriptions or rationale for this concept.

An example of the above is how Magpie uses lexicons. A lexicon is a serialized transformation of a complex populated ontology for a particular purpose. In Magpie, lexicons contain also domain-specific and generic lexical/syntactic variants of the semantic objects, which in turn enables (i) the association of concepts with entities in a web page, and (ii) the significant increase in the speed of the initial mark-up of a web page. Obviously, in other applications, other views might be more suitable.

One topic that is gaining popularity at the time of writing this article is the support for facilitating views on ontologies via mapping or aligning them [23] with each other or with other established descriptive structures such as glossaries or taxonomies. The need for aligning ontologies is becoming increasingly urgent to tackle heterogeneity, but we decided not to include this requirement in an explicit form here. The main rationale is that as the alignment methods mature, they are likely to lead to establishing alignment servers[1] to which individual Semantic Web applications may simply refer, rather than carry out alignments on their own. Thus, the key remains the capability to take knowledge from multiple ontological sources.

3 *It has to offer some semantic services to operationalize the populated ontological model*.

Having provided the user with a choice of ontologies and of some views on the populated knowledge bases,

[1] An example is Euzenat's alignment server at INRIA (http://alignapi.gforge.inria.fr/).

a Semantic Web application should be able to carry out inferences related to a user's need. Thus, the mere presentation of a marked-up item in a Semantic Web language (such as OWL), however stylized, is not sufficient to make a tool into a Semantic Web application. Ideally, Semantic Web applications should (i) make use of Semantic Web technologies, such as mark-up languages, and (ii) provide means to execute inferences about items annotated with such technologies.

Furthermore, this criterion draws attention to the aspect of "service"; in other words, the inference actions shall be open and reusable to the widest possible extent. As a side-effect, this criterion suggests treating a Semantic Web tool as a modular, extensible application to which new functionalities could be easily added by referring to a new service.

4 *It has to be able to translate from the semantic to "non-semantic" user interaction means and modalities*.

Finally, we have included this criterion to bring in the human user of Semantic Web applications, and also to comply with Quan and Karger's point on separating content from presentation [44]. Why are we distinguishing "non-semantic" user interactions? HCI researchers have long seen user interfaces as means to convey some metaphors to the users, where metaphors are seen as references to *familiar* habits, tasks or concrete objects (see, e.g. [38]). Familiar, concrete metaphors enable users to work with abstract concepts and simplify the construction of a "how things work" model or a "user illusion" of a computational tool [35].

Due to young age of semantic technologies and their close relationship to more generic web technologies, our users have developed models coping with the web content and derived them from the traditional metaphors like books, bookmarks, point & click commands, etc. While these metaphors might not be ideal for Semantic Web, according to [41], "successful interface metaphors draw heavily on the user's knowledge of the world around them, and on established conventions that allow the user to predict the results of their actions in advance". Hence, established metaphors, though imperfect, are notoriously hard to give up.

This criterion is not about some specific "semantic interaction" that needs to be implemented. When we refer to "non-semantic interactions" we mean standard, familiar user interaction metaphors that may be also applicable to Semantic Web data. Hence, one purpose of translating semantic knowledge into more familiar non-semantic user interaction metaphors is to enable the users accustomed to a range of desktop and web interfaces to start using Semantic Web applications without incurring too high a cognitive overhead. In fact, the user may not even be interested in the question whether s/he uses any semantics in his or her browsing.

Such a translation may at its simplest involve the use of style sheets, or, in its more complex form, could disguise the semantic commitments and inferences behind familiar user interaction metaphors such as colored highlights or dynamically created contextual menus—as shown in the examples of Magpie application from Section 2.

3.3. Discussion of the proposed criteria

Comparing our four criteria on the Semantic Web application with the three evolutionary steps of the Magpie technology, it could be argued that the first generation of tools was more akin to mock-ups using some aspects of Semantic Web technologies rather than real Semantic Web applications. From our four criteria we see the dominant role played by *multiple ontologies* as the defining feature of a Semantic Web application that may not be satisfied by some popular tools that use semantic mark-up but do not commit to any particular ontology (e.g. various tagging systems such as Flickr[2]).

Since the first applications of Semantic Web technologies in early 2000s much effort has been put into criterion 3—development and distribution of functional capabilities via (semantic) web services. Many new applications expose their data via an open service-based interface. On the other hand, criterion 4 seems to be lagging. While XML style sheet formalisms apply also to Semantic Web languages, a more generic issue of how a human user can interact with the Semantic Web content is only now becoming a fully-fledged research problem.

Indeed, the shortcomings of many current Semantic Web tools are remarkably similar to the problems with social tools in general that were noted a decade ago by Grudin [28]. In addition to already mentioned technology adoption and immediate rewards for the users, other challenges in Grudin's analysis include creating parity between user effort and benefit, achieving a critical mass of users and resources, and developing tools that are unobtrusive and accessible. *Although Grudin's original study was about social and collaborative tools, many of his challenges still (or again) apply to the research into tools and technologies for the Semantic Web*.

For instance, Grudin's unobtrusiveness could be mapped onto our criterion 4—making Semantic Web technologies look more familiar to the users. Similarly, the notions of rewards for the user and technology adoption are visible in our criteria 2 and 4—in addition to low-level formal representational languages using a range of views on the marked-up data. Indeed, the latest developments of the tools showcasing Semantic Web ideas are following precisely in this direction—proving that the design requirements we had in the original Magpie publications [18–20] were indeed correct. For example, the latest instantiations of PiggyBank [32] (combined with Thresher [31] as a Magpie-like user frontend) or AKTive Document [36] (combined with a range of text processing techniques developed by the same team from the University of Sheffield) do indeed combine the knowledge acquisition functionality (via screen scraping or entity recognition) with a subsequent presentation of this acquired knowledge *in situ*; i.e. in the context of a web page they were found on.

One notion we have not included in our list of requirements is the scalability of the applications in terms of coping with a large number of rich knowledge-level statements. Although Grudin has recognized the issue of a critical size (of data or user base) in computer applications, we believe that an added value of

[2] Flickr (http://www.flickr.com) is a registered trademark of Yahoo!

Semantic Web applications could be in improving the quality of user support. Moreover, the scale on the level of data repositories is fairly well supported by databases and RDF stores.

To conclude this discursive reflection on the evolution of Magpie, we believe the four criteria presented above may serve as lessons learnt from our experiments with various generations of the Magpie technology. It is not our intention to judge which application should be called "semantic" and which should not. On the contrary, the value of our reflection is in the potential contribution to a design of new, emerging applications—applications that would work, e.g. with multiple, larger ontologies, with context-specific views on the ontologies, or with dynamically orchestrated services.

In the next section we illustrate how the issues and criteria discussed above relate to the usability of Magpie-based semantic applications, and how they have driven the evolution of the framework through the three stages mentioned earlier – from the creation of semantic layers, through Semantic Web browsing to the framework for developing larger semantic applications – towards the four generalized criteria.

4. Beyond handcrafted Semantic Web browsing

To illustrate how we responded to the evolving requirements Semantic Web technologies need to satisfy in users' tools, let us return to the Magpie-based applications. One of the prominent issues driving Magpie improvements was the requirement of lightweight, *usable tools for end users that automate some complex task*. The key evolutionary driver has been the need to tackle seemingly independent challenges: (i) designing user interfaces that are sufficiently robust yet simple, (ii) supporting automated ontology population and the subsequent automated generation of reliable Magpie lexicons, and (iii) integrating knowledge maintenance with the Magpie tool and framework.

4.1. Evolution driver: evaluations of Magpie-based applications

First, from a *handcrafted solution* for finding and displaying semantic annotations in a web browser with a fixed set of actions [20], Magpie moved to an *open solution* with dynamically definable actions for navigating the Semantic Web [21]. The climate science application reviewed in Section 2 fell short of fulfilling some of the pragmatic needs; for example, supporting knowledge acquisition and evolution. The ontology used in the climate science application of the Magpie framework was manually crafted. The knowledge base (KB) was populated by mining the Web, but the population was a one-off rather than a continuous exercise. And finally, the climate science application was not fully adaptable; for instance, in terms of learning new knowledge, using new services, or generally, maintaining existing knowledge.

Like Magpie, other applications from the early period of Semantic Web research fall well short of addressing multiple stages of the knowledge flow through its lifecycle. For instance, Haystack [43] features knowledge reuse and sharing but has limited support for automated acquisition and maintenance. Protégé [26] focuses on the representation with basic versioning/mapping support. GATE [13] and KIM [42] support discovery and semantic annotation, but do not support the reuse and maintenance of the discovered knowledge. SCORE [47] or UIMA [25] have powerful information extraction engines and introduce knowledge maintenance and semantic search, but no additional services. Also, they are frameworks for developers with no direct support for user interaction aspects. Annotea [33] is user-centered and derives its functionality from the actual tasks of a specific group of users, but its free-text annotations are more suitable for informal bookmarks/notes than for automatically discovering ontologically bound annotations. Web browsers in general are good at document presentation but do not have any knowledge discovery, creation and maintenance functionalities.

Table 1
Summary outcomes of evaluating the benefits of automated knowledge acquisition and ontology population added to the Magpie framework (see also [49])

Data gathering task	Group A	Group B	Group C
People	13.2	15.3	13.7
Technologies	19.2	23.4	26.7

In order to interact with the Semantic Web the user needs a toolkit that efficiently connects semantic and standard (i.e. non-semantic) browsing techniques; e.g. using the automated semantic annotation of web pages [49]. However, the experience with Magpie also shows that the automated recognition of terms and their annotation is often brittle [49]. Although Magpie was never positioned as a dedicated language processing tool, it can be evaluated using standard measures of language processing, such as precision of concept recognition and rate of recall. Term recognition in Magpie occurs in real time and with a high precision while the resource is *within* the selected ontological domain. As expected with this approach, if a Magpie lexicon is used on the text outside of the domain of a user-selected lexicon, performance declines.

Hence, we wanted to assess how good Magpie framework might be in meeting the bootstrapping challenge: the automation of the resource annotation process, as proposed in [47]. We also wanted to investigate to what extent our theoretical arguments of supporting knowledge lifecycle were true. We compared the variance in performance between the default, brittle approach to entity annotation using a manually crafted lexicon with an approach based on building lexicons using information extraction techniques. If successful, this would allow more automation and consequently, the production of *more robust browsers for the Semantic Web*, as we conceptually suggested earlier, in Section 3.

When we compared the brittle Magpie lexicons with those augmented by the information extraction tools eSpotter [49] and PANKOW [11] (see [49] for the details of the evaluation study and analysis), the users' performance in an exploratory task indeed improved when using the augmented lexicons; thus, making the domain boundaries less brittle. The improvement in user performance when attending to additional phases of knowledge life cycle is visible from the rise in means shown in Table 1. However, this achievement came at the price of not being able to generate lexicons in real time—the discovery of additional knowledge takes time.

Table 1 summarizes the average number of semantic entities discovered by the participants in our ethnographic study using Magpie in a data gathering task focusing on finding people and technologies in a given corpus of web pages. The findings had to satisfy a number of given criteria (e.g. distinguishing visitors from the staff) without any prior knowledge. The column labeled as “Group B” includes participants using a Magpie-equipped web browser with lexicon derived from a handcrafted database and augmented by eSpotter technique. “Group C” includes the same lexicon augmented by the PANKOW techniques, and finally, “Group A” represents a control group (no automatic augmentations of lexicons used).

As Table 1 shows, participants’ performance and recall between groups A and B improved by 15–21% in both studied tasks; simply by allowing minor augmentation to the handcrafted lexicon. However, when we compare Group B and Group C for gathering people task (middle row), there is a fall in recall. We would interpret this as a situation where the fully automated discovery and annotation of entities was inconsistent with the pragmatic challenge of real-time responsiveness. The users found it more difficult to process larger number of semantic annotations in the same text, within the same time interval. Thus, this study has supported our conceptual proposition of a primary source of tension in the Semantic Web. In Section 3 we introduced this tension in terms of balancing the automation of the bootstrapping process and the perceived reward for the users.

Another issue arose with the breadth of opening up well-defined ontologies for automated population. If too many diverse items were discovered for the users, this may improve the knowledge acquisition phase and give them more candidates to process. However, a larger number of candidate entities may, in some cases, also *reduce users’ satisfaction* and indeed performance. This aspect is documented in the “rise then fall” effect on the people task, as shown in the upper row of Table 1.

4.2. *Lessons learnt from the evaluated Magpie applications*

The third viewpoint on Magpie, as introduced in Section 3.1, is very helpful to address the tension we empirically found in our study. The key idea is to treat the Magpie framework (and possibly other Semantic Web applications) as “shells” that provide generic mechanisms for integrating the ontologies, knowledge bases (KB) and web accessible resources with hyperlinks (semantic) web services and tools interacting with them. Rather than producing applications for merely browsing the Semantic Web, we proposed to view them as *integrated solutions for managing knowledge* (*on the Semantic Web*). In other words, Magpie’s open service framework enables specialized services (e.g. population, validation, etc.) to be linked.

This view distinguishes between partial solutions to specific sub-problems (say, *Semantic Web techniques*) and more integrated user-centric *Semantic Web tools* and applications. Thus, in general, a successful Semantic Web application would draw upon a range of specialized techniques to enhance its core functionality. Once again, an integrated solution in this sense is in contrast to a monolithic solution, where an application would be expected to take care of every aspect alone. In an integrated solution, an application merely refers to and invokes a specialist technique or service, if this is beneficial to the users and their experience with the application.

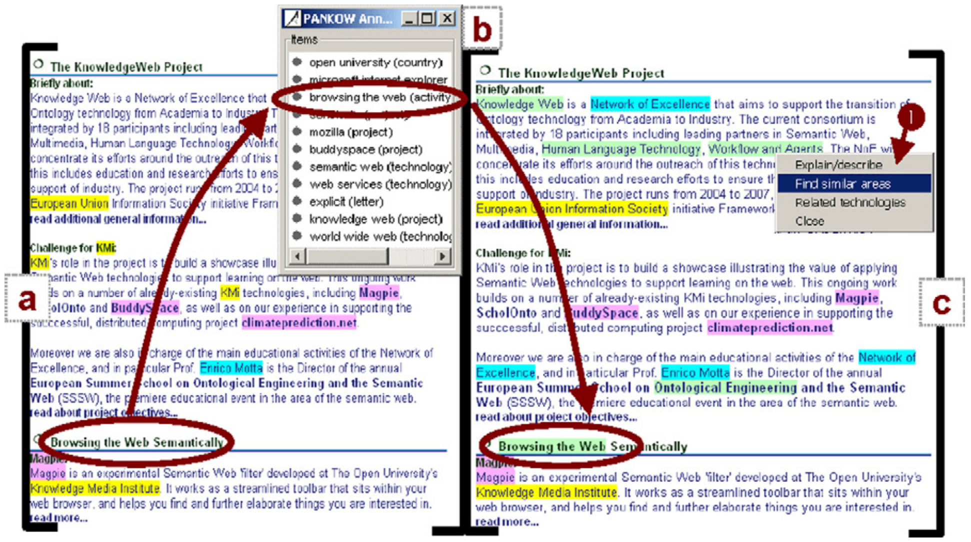
Linking the specialized services broadens the scope of the semantic application from supporting one stage in the knowledge life cycle (e.g. knowledge presentation via web browsers) to supporting the knowledge life cycle itself. Thus, merely automating the knowledge acquisition may not be beneficial to the user in all circumstances, as Table 1 shows. Acquisition is only one activity that needs to be complemented by others. Magpie supports these complementarities by allowing its semantic services being linked in two ways. First, one can aggregate service outcomes at the application’s front-end. This was mentioned in Section 2 when a service “*Explain concept*” was able to offer graphical output if any was available. However, this linking of services can also be performed in the application’s background; e.g. when generating new lexicons from automatically updated knowledge bases.

Thus, the main lesson learned from developing and deploying semantic applications based on the Magpie framework is encapsulated in the need to cope with the open space of the Web. Consequently, Magpie evolved from its original role of a viewer/navigator with a predefined viewpoint more towards an originator and driver of the knowledge acquisition and maintenance processes. The vision of embedding a semantic application in a broader context of networked, web-based knowledge space is novel, in our opinion, and aims to balance the pragmatic, real-time browsing (or more generally, user interaction) requirements with the requirements on bootstrapping and knowledge evolution.

In the next section we give further details of Magpie in the role of an open Semantic Web framework that is capable to carry out some aspects of knowledge maintenance. We argue that integration using the Magpie framework addresses the needs of both the end users (with their interest in browsing, retrieving and annotating), and the developers (who need to create and maintain KBs). As such, this vision is an intersection of three areas that are currently perceived as separate: Semantic Web, web-based services and knowledge management.

4.3. *Magpie as an integrated solution for semantic knowledge maintenance*

We already mentioned that the adoption of the Semantic Web depends on matching different needs of different users with the automated means of knowledge creation. Focusing solely on end users is rewarding in the short term, but in addition to the instant rewards, semantic applications should offer sustainable, delayed gratification [48]. In terms of functionality this means that a tool for the Semantic Web (obviously) needs to address the requirements of its end users. From the perspective of users, such a gratification is in making the browsing on the Semantic Web more robust, more effective, and more adaptable—in short, more instantly rewarding [39].



The KnowledgeWeb Project

Briefly about:
Knowledge Web is a Network of Excellence that aims to support the transition of Ontology technology from Academia to Industry. The current consortium is integrated by 18 participants including leading partners in Semantic Web, Multimedia, Human Language Technology, Workflow and Agents. The NoE wi concentrate its efforts around the outreach of this techn this includes education and research efforts to ensure th support of industry. The project runs from 2004 to 2007, European Union Information Society initiative Framewor
read additional general information...

Challenge for KMi:
KMi's role in the project is to build a showcase illustrating the value of applying Semantic Web technologies to support learning on the web. This ongoing work builds on a number of already-existing KMi technologies, including Magpie, ScholOnto and BuddySpace, as well as on our experience in supporting the successful, distributed computing project climateprediction.net.

Moreover we are also in charge of the main educational activities of the Network of Excellence, and in particular Prof. Enrico Motta is the Director of the annual European Summer School on Ontological Engineering and the Semantic Web (SSSW), the premiere educational event in the area of the semantic web.
read about project objectives...

Browsing the Web Semantically

Magpie:
Magpie is an experimental Semantic Web 'filter' developed at The Open University's Knowledge Media Institute. It works as a streamlined toolbar that sits within your web browser, and helps you find and further elaborate things you are interested in.
read more...

Fig. 3. Extract from a web page annotated with a brittle KB and lexicon (a); a list of KB extensions proposed by C-PANKOW and visualized in Magpie collector (b); and the same page annotated by an extended KB/lexicon as learned by C-PANKOW and validated by Armadillo (c).

Simultaneously, a semantic application may need to exhibit features addressing the requirements of knowledge engineers, librarians or managers. From their perspective, the requirement is to increase the automation and reliability of knowledge acquisition, and reduce the overall complexity of managing semantic resources, knowledge bases, ontologies, annotations, etc.—in short, making the application rewarding in the longer term and thus sustainable [48].

A dedicated Magpie-based application was developed to test the integration of specialized services to match the needs of different users as suggested above. Similarly as the application reviewed in Section 2, this prototype supported e-learning; namely in the domain of Semantic Web Studies. The KB for this domain had been acquired automatically, by scraping a small corpus of web-based resources and database entries that were considered to be relevant by the application design team. This initial set of concepts and instances was insufficient to deliver a high-quality user experience, and additional items had to be acquired dynamically.

The transition from the initial, brittle lexicon towards an extended one acquired with a help of web mining tools is illustrated in Fig. 3: in the sequence of screenshots from (a) the annotation using a brittle lexicon, (b) the recognition of additional items in a visited web page, to (c) the page annotated with an extended lexicon.

On the left, Fig. 3a shows a web page extract annotated using an initial lexicon populated from an internal database. As can be expected, concepts like "*Magpie*" or "*BuddySpace*" are highlighted because they were explicitly defined; however, a few generic but conceptually relevant concepts are ignored (e.g. "*web services*" or "*human language technology*"). These may be related to the existing terms but are not in the scraped KB. This is due to (i) incomplete knowledge acquisition and (ii) ontology brittleness (i.e. the rapid degradation of performance in an open problem space).

To overcome brittleness, Fig. 3b shows a collection of additional, potentially relevant instances discovered by an information extraction tool that was linked through the Magpie's trigger service function. Trigger services have been discussed in more depth elsewhere [19], so we focus here on the functional advances of this particular design rather than technicalities. Newly discovered entities were related to the domain of the original, user-selected lexicon; yet did not exist in the KB, from which annotation lexicons were generated. The discoveries were on the level of instances, but the IE tool also proposed their coarse-grained classification using classes that are already known in this particular domain ontology; such as "*Activity*" or "*Technology*".

The results of IE are shown in Magpie's dedicated interfaces, called *collectors*, which visualize the responses of all trigger services in Magpie. A trigger service is not invoked by the user directly, but rather responds to a particular pattern in the web pages the user visited. In Fig. 3b, an external IE engine processes each web page visited by the user, and newly discovered concepts or instances are "pushed" back to the user and appear as a linear collection of terms and their categorizations.

At any time, the user may have several collectors activated. This gives the user control over the amount of new information being presented, and it is intended to manage the information load on the user. In addition to merely aggregating items, Magpie collectors offer a range of other functions, such as semantic bookmarking or browsing history management. These are discussed together with trigger service and semantic logging functionality in [19].

Finally, Fig. 3c shows new instances (e.g. "*browsing the web*" or "*workflow and agents*") already incorporated into the KB and used for annotation. Importantly, the newly discovered entities are not only highlighted as "*Research activities*", but the user can also invoke associated services to obtain additional knowledge about these discoveries. The semantic menu marked as ❶ in Fig. 3c shows how the user is selecting one of the services for the newly discovered instance.

The main challenge from a knowledge engineer's perspective is to avoid having the users manually committing the proposed discoveries into a KB. Such revisions are, admittedly, not what

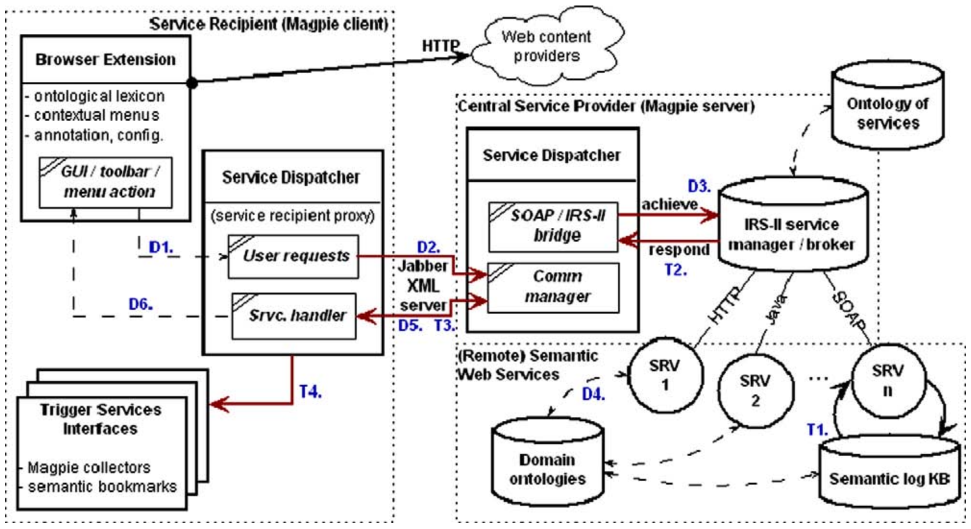


Fig. 4. Schematic architecture of "open services" Magpie framework.

most users would have the privilege, expertise or will to do. Extending Magpie lexicons by applying IE techniques provides benefits to the end user by making their Semantic Web browsing more robust and less brittle, but these benefits do not extend to knowledge engineers. Provided a *small number* of concepts are discovered and only *occasional* amendments are needed, manual revisions of the KB-s are possible, but they are not scalable.

In our experiments with the IE-extended Magpie, the knowledge engineer had to manually adjust the classification for 21–35% of discoveries in each category. This was still a substantial effort in terms of maintaining knowledge. However, by linking one IE engine to a validating IE technique, we immediately reduced the need for adjustment (e.g. for *Events* down to 8%). Thus, by integrating the pragmatic [6] IE techniques and by making them accessible from a semantic browser we achieved benefits for developers and other KB users—in terms of evolving knowledge and assuring its quality.

Benefits for the end user centre on more robust browsing, but also provide opportunity to personalize a generic, handcrafted KB for a particular user or group. While the personalization was not the focus of our case study, it is an important side-effect of our aim to develop semantic applications addressing not only the annotation and presentation of knowledge, but also other stages of the knowledge processing chain (e.g. knowledge re-use and maintenance). Once the integrated application is capable of creating knowledge, this capability may be applied to personalizing KBs. Personalization, in turn, may lead to creating tools that are aware of pragmatic issues, such as trust or provenance [34].

5. An open semantic services architecture

Drawing on our previous publications (esp. [21]) we now summarize the basic architecture that underlies all Magpie applications. The architecture (shown in Fig. 4) has been designed to allow Magpie users to define, publish and use their own semantic services. It is based on an infrastructure we have developed in the past—IRS-II [40], which supports the publishing and invocation of semantic services. IRS-II is based on the UPML framework [24], and therefore differentiates between *tasks*, *problem solving methods* (generic reasoners) and *domain models*.[3] By distinguishing between tasks and problem solving methods we effectively separate the activity of specifying and implementing semantic services (problem solving methods) from making their descriptions available in a form that is more familiar to a user (tasks).

The details of the Magpie architecture have been discussed from different angles in [18,20,21], therefore here we only provide a brief overview of its two core modules: a *service provider* and a *service recipient*.

5.1. Service recipient components

On the service recipient side the framework features: the *Magpie Browser Extension* (plug-in of Internet Explorer or Mozilla), the *Magpie Client-Side Service Dispatcher*, and *Trigger Service Interfaces*. The first has been described in [19,20], and also its functionality was illustrated in Section 2; the other two aspects appeared in [19]. In a nutshell, the basic Magpie plug-in provides the capability of automatically matching items in a web page to instances of the selected lexicon (a serialized ontology) and the capability of managing right-click menus with contextualized semantic services.

The Magpie Client-Side Service Dispatcher acts as a dedicated proxy for the communication between the service providers, the Magpie-enabled browser and the user. In principle, it is an alternative to the GET/POST requests available for standard hypermedia. Although Magpie supports such requests, a growing number of services are available in formats not suitable for integration with a standard web browser, and for this reason the Magpie architecture supports a more generic approach to service invocation. The Dispatcher delivers both

[3] IRS-II became in the meantime obsolete and was replaced by WSMO-compatible service execution environments, e.g. IRS-III [8].

user requests and the responses from providers encoded using XML-based envelopes [4]; e.g. to be used by collectors.

The Magpie Dispatcher acts on behalf of the user and can be identified as such. Hence, because the service provider is aware of the user's identity, it is possible to communicate service requests/responses *asynchronously*. This is an important extension of the standard hypermedia protocol, which assumes synchronous, stateless interactions. One situation where this comes useful is collectors mentioned in the previous section. The collectors are a form of Magpie Trigger Service Interfaces, which visualize the XML-encoded data pushed by the specific trigger services a user subscribes to. Trigger services (mentioned in Section 4.2) are an innovative feature, which, unlike the menu-based services, gets activated by patterns among entities recognized in the web page and automatically asserted in a semantic log. A log is basically a simple, transient KB maintained for a user or a group of users storing various assertions about the discovered entities.

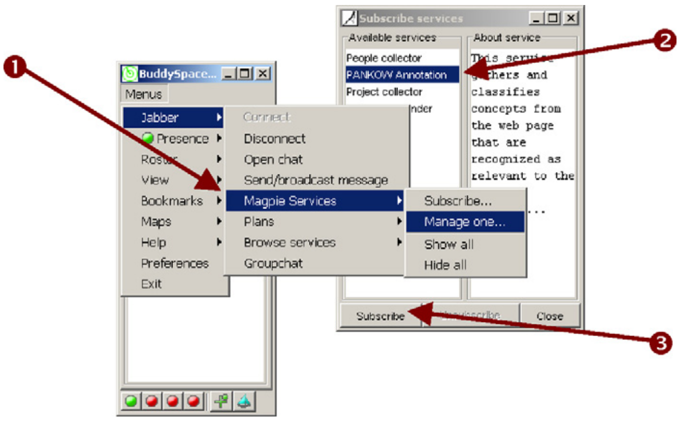
Support for asynchronous interaction between the client and the server makes the Magpie architecture extremely flexible. The bi-directional information exchange may, in principle, support tasks such as negotiation or dynamic update of lexicons. It may also facilitate simple personalization of lexicons or of responses to certain semantic services. For instance, different degrees of response granularity may be available, or ontologies may be stored in different formats; the choice would be made automatically based on a user's privileges or preferences. The dynamic aspects become particularly useful in scenarios as described in Section 4.2, when we have an entity extraction service attached to the generic Magpie plug-in for knowledge maintenance.

5.2. *Integrating third-party services with Magpie*

Earlier we mentioned that lexicon-driven text processing and semantic enrichment techniques suffer from *brittleness*. A capability to discover new knowledge in plain text is thus important to address the knowledge (and ontology) maintenance part of the Semantic Web challenge. The Magpie framework facilitates plugging in third-party text processing methods to implement such a capability. However, the more precise a text processor is, the greater the time overheads. In this situation, Magpie's asynchronous communication strategy comes particularly useful.

5.2.1. *Subscribing to a third-party information extraction service*

As mentioned in Section 4.3, we experimented with C-PANKOW [10] as an IE engine recognizing instances related to a specified ontology in text. There now exist many IE tools, and some are more powerful than C-PANKOW. For example, techniques forming IBM's UIMA architecture [25] have functionalities for identifying relations and entity co-references in addition to basic entity recognition. However, our primary objective in this experiment was to test that multiple service extensions would work together for the user's benefit. The purpose of an additional IE technique was *not to achieve superiority in a single phase of knowledge processing* chain but rather to verify the *feasibility of linking separate techniques that specialize* in one task



This service gathers and classifies concepts from the web page that are recognized as relevant to the ...

Fig. 5. User's subscription to a third-party text processing service/tool.

(e.g. co-reference resolution), which is a common real-world problem solving tactic.

The experimental C-PANKOW was published by its developers as a service within the Magpie framework, and it became available for user subscription using the Magpie Dispatcher module. The subscription to and activation of C-PANKOW by a user is shown by marker ❶ in Fig. 5. Marker ❷ shows the list of services available to a given user. By subscribing to a selected service (see marker ❸), the user activates its triggering mechanism and also a respective Magpie collector.

5.2.2. *Receiving notifications from an IE service*

Provided the C-PANKOW service is active, upon each visit to a new web page the Magpie plug-in sends a notification to the alert management task. The alerts are re-distributed to appropriate trigger services the user was subscribed to. In the case of C-PANKOW, the URI of the page the user visited together with the selected domain ontology and user's identity were used as inputs for starting IE on a remote computer.

C-PANKOW algorithm [10] took up to 3 min to process each page. After processing, the C-PANKOW service invoked a messaging service of the Magpie framework to deliver the discovered items to the user. An example of responses received by the user is shown in Fig. 6. Each of the items (see, e.g. marker ❹ in Fig. 6) is assigned to a class from the same ontology that was used by the user to initialize the Magpie plug-in.

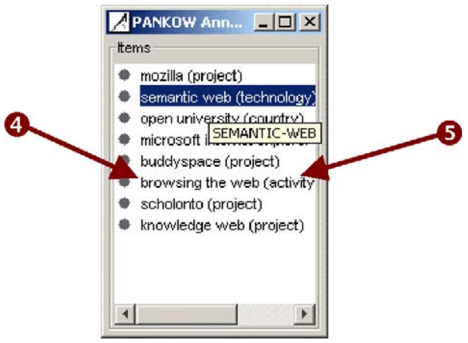


Fig. 6. Outcomes of the third-party NER service being visualized in the Magpie collector (after being communicated asynchronously).

One can now explore how to improve both the efficiency and the effectiveness of C-PANKOW in its role of a knowledge maintenance service. However, we chose a different experiment instead: direct delivery of the discovered knowledge to the user was replaced by another technique that acted as a knowledge validator. For this purpose we used Armadillo [12]—a tool co-developed with C-PANKOW but with a different IE strategy. Armadillo took as an input the entities recognized in the first pass by C-PANKOW and checked them for co-references using additional web resources relying on the principle of information redundancy [12].

This enabled the chain of originally independent *techniques* (Magpie, C-PANKOW and Armadillo) to combine into a larger *application* centered around a particular ontology. Again, it would be possible to use a more powerful IE engine or architecture such as UIMA that addresses the issue of validating the extracted data. However, the objective of *this* experiment was to assess the claim that an end user centered application for the Semantic Web (such as Magpie) could be seen as an integrated solution consisting of *a range of ready-made specialized techniques*. Through aggregation applications become less brittle and performing better in the open and distributed environment of the Web [46].

5.3. Service provider components

A number of components on the service provider side manage value-added functionalities such as semantic logging, the association of semantic services with ontological classes, the invocation of semantic services selected by the user, and reasoning for trigger services. Two criteria for designing this "back-office support" for Magpie are:

- Openness to the definition of new semantic services, which may use an existing ontology and a require access to the semantic log.
- Allowing users to customize how the output of a service is rendered.

As already mentioned, we have fulfilled this requirement by integrating Magpie with the IRS-II architecture [40]. The Magpie Service Manager used the IRS-II framework to handle the requests of the individual users for the individual services and to communicate with the IRS broker, whose job is to locate the appropriate web service when a request is made to achieve a particular task (e.g. "*Explain concept*", which was used as an example in Section 2).

Conceptually, the integration between Magpie and the IRS is achieved by defining a top-level *Magpie task,* which takes as input an ontological identifier of the clicked-on entity, a set of related arguments and a choice of a visual renderer specifying the desirable output format (the default is HTML). All semantic services inherit from this generic task, and extend it with specific input or output roles (e.g. the semantic service "*Explain concept*" described in Section 2 takes a given instance as input and offers a textual definition or a pair of textual/graphical definitions as its output).

A task can be handled by one or more *problem-solving methods* (PSMs). PSMs are the knowledge-level descriptions of code for reasoning with particular input roles as specified by the task definition. PSMs introduce a system of *pre-conditions* (e.g. an argument supplied should not be an instance of "Physics" or "Chemistry"), and *post-conditions* (e.g. refer to graphics only if available). While PSMs are crucial for reasoning, the end-user only interacts on the task level—thus specifying what needs to be achieved rather than how to do it. The IRS-II and IRS-III frameworks support different modes of service publishing and invocation [8,40]; including "via URI" which are used in Magpie web browser plug-ins.

The process of adding a semantic service to the Magpie application is a custom variation of a generic semantic service publishing, and has been explained in [40]. Further details on how an individual service can be created from a standalone web application; including examples were published earlier; e.g. in [21].

6. Related work

A tool that functionally resembles Magpie is the KIM plug-in for Internet Explorer. Knowledge and Information Management (KIM) is a platform for automatic semantic annotation, web page indexing and retrieval [42]. As Magpie, it uses named entities as a foundation for formulating semantic relationships in a document, and assigns ontological definitions to the entities in the text. The platform uses a massive populated ontology of common upper-level concepts (e.g. locations, organizations, dates or money) and their instances.

Unlike Magpie, KIM is based on the GATE platform [13] but it extends its flat entity recognition rules with an ontological taxonomy. The entities are recognized by the KIM proxy server, and, in parallel, are associated with respective instances in the ontology. GATE supports the recognition of acronyms, incomplete names and co-references. This enables KIM to work with both already-known and new entities, which contributes towards a knowledge maintenance capability.

Magpie differs from KIM in a number of respects. While KIM is coupled with a specific, large knowledge base, Magpie is open with respect to ontologies, allowing users to select a particular semantic viewpoint and use this to enrich the browsing experience. Another difference is that while KIM is very much steeped in the classic "click&go" hypermedia paradigm, Magpie is open with respect to services, as shown by examples in this article. As already pointed out Magpie goes beyond KIM in the direction of providing a framework for building Semantic Web applications, rather than simply supporting entity recognition and semantic annotation.

Magpie also differs from "free-text" document annotation tools [29,33] by intertwining entity recognition, annotation and ontological reasoning. Annotation based on ontological lexicons outperforms "free-text" annotations in terms of a greater than 90–95% recall rate and a similar precision score for in-domain resources. Yet, free-hand annotations are useful for an ad hoc, personal, customized interpretation of the web resources. Magpie does not currently support manual semantic annotation, which is a limitation.

Another system superficially similar to Magpie is COHSE, which implements an open hypermedia approach [9]. The similarity between the two systems is due to the fact that (at a basic level of analysis) both work with web resources and use a similar user interaction paradigms (“click&go”). However, beyond the superficial similarity there are major differences between these two systems. The main goal of COHSE is to provide dynamic linking between documents—i.e. the basic unit of information for COHSE is a document. Dynamic linking is achieved by using the ontology as a mediator between terms appearing in two different documents. COHSE uses a hypertext paradigm to cross-link documents through static and dynamic anchors.

In contrast with COHSE, Magpie is not about linking documents. In Section 3 we discussed three ways of looking at Magpie: as a tool to support the interpretation of web resources through “ontological lenses”, as a way to support Semantic Web browsing, and as a framework for building Semantic Web applications. In particular, in the latter perspective, Magpie goes beyond the notion of hypermedia systems, by providing a platform for integrating semantic services into the browsing experience, both in pull and push modalities.

In a nutshell, Magpie uses a different paradigm that sees the Web as an environment for knowledge-based servicing of various user needs. Using a “Web as computation” paradigm, we not only provide information about one concept, but can also offer knowledge dependent on *N-ary relationships* among concepts. For example, this can be seen in the context of semantic collectors (examples in Section 4.3): since these take in account patterns among the concepts within web pages, the patterns would usually involve more than one concept. An example of a scenario where a more complex relationship was applied involved a collaborative work [18]. This is rather difficult to achieve in a hypermedia system—one cannot use one click to follow multiple anchors simultaneously and reach a single target document or a piece of knowledge.

Moreover, Magpie supports the publishing of new services without altering the servers or the plug-in. Service publishing also enables the users to subscribe to selected services. It also makes the development of a semantically rich application more modular and thus cheaper and easier for domain experts rather than knowledge engineers. This is more powerful than the mechanisms used by open hypermedia systems, which are largely based on the editorial choice of links. Magpie explores the actual knowledge space as contrasted with navigating through hypertext (which is one explicit manifestation of a knowledge space). Mere link following (in open or closed hypermedia) is not sufficient to facilitate document interpretation. We complement the familiar “click&go” model by two new models: (i) “publish&subscribe” (for services) and (ii) “subscribe&acquire” (for data and knowledge).

Yet another approach to associating meaning with content has been developed in the Semantic Content Organization and Retrieval Engine [47]. SCORE is conceptually closest to the Magpie’s role as a framework for developing Semantic Web applications. The main difference from the core Magpie is the reliance on regular expression rules for IE instead of an ontology-derived glossary or gazetteer. Ontologies are used to index and classify extracted information. The purpose of SCORE is then to facilitate semantically grounded search in a collection of annotated documents. SCORE also aims to support what can be labelled as contextually broadened search and the retrieval of marked-up information, which differs from Magpie’s emphasis on the interpretation of the content and the exploration of the knowledge space rather than search. SCORE also emphasizes the IE task, unlike Magpie that aims to take in account user interaction and user experience aspects.

From a user interface adaptability perspective Magpie is relevant to projects such as Letizia [37] with its reconnaissance agents. This type of agent “looks ahead of the user on the links on the current web page”. Such pre-filtering may use semantic knowledge to improve the relevance and usefulness of browsing. Magpie implements a functionality similar to that of Letizia (“logged entities reconnaissance”) through semantic logging and trigger services, and thus provides a more general and flexible framework for implementing push services, than the one provided by Letizia.

To conclude we want to note a growing recognition by the research community of the need to make the Semantic Web accessible to “ordinary” users. Many approaches now follow similar, lightweight and near-zero overhead principles as Magpie; albeit for different purposes. The authors of Haystack [43] and Mangrove [39] see the major issue with Semantic Web in the gap between the power of authoring languages such as RDF(S) or OWL and the sheer simplicity of HTML. In response to this concern, Magpie separates the presentation of semantic knowledge, service publishing and invoking from the underlying knowledge-level reasoning mechanisms.

Since the development of SCORE, Magpie, Haystack or Mangrove, new tools have emerged following the requirements we introduced in connection with our Magpie research in 2003. Most recent releases include Piggy-Bank [32] and AKTive Document [36]—both re-visiting Magpie’s demand of re-using familiar interfaces, lightweight processing, immediate rewards and user friendliness. Both tools extend the original Magpie in terms of greater support for interaction with the user. The user is no longer restricted to be in a position of inquirer or viewer. Both, Piggy-Bank and AKTive Document give users opportunities to create, share, import and re-use annotations from multiple sources (e.g. other people’s bookmarks in Piggy-Bank). This is useful because it facilitates manual knowledge acquisition from the users, which in turn contributes to tackling the pragmatic issue of evolving knowledge in an open environment of the Web, which we mentioned earlier.

The notion of ontological perspective selectable by the user that was one of the original Magpie innovations, also found popularity. For instance, VIeWs [7] enables visitors, who come to an information portal, to choose between several, ontology-grounded perspectives, which, in turn, inform what kind of knowledge will be made available to them through semantic services menu. VIeWs emphasizes knowledge customization for different audiences—e.g. tourists vs. business visitors to a region. This has been mentioned earlier, too. The capability to customize interaction with semantic knowledge is one of the key benefits one can derive from the Semantic Web.

These tools can be seen as a natural extension of the research into semantic tools à-la Magpie—tools that take great care of bootstrapping the Semantic Web through making user experiences more rewarding. From this perspective, Magpie's original design requirements formulated back in 2002 [17], which emphasize the concepts of "zero overhead", real-time responsiveness, flexibility and re-use of familiar user interfaces, have, to some extent, informed a range of valuable research experiments and prototyping of user interfaces and user interaction on the Semantic Web.

7. Conclusions

In this article we described the Magpie framework and its evolution through three phases—from seeing it as a single-purpose tool supporting the interpretation of web pages, through a generalized prototype of a Semantic Web browser, to an open, service-based architecture for developing Semantic Web applications. We have linked these evolutionary developments to the broader issues and challenges for the Semantic Web research community.

We have drawn several lessons from developing and deploying Magpie-based applications. First, Magpie has proven useful with respect to several objectives. Being based on the familiar interaction metaphor of a web browser, it featured a friendly way for the ordinary user to perceive the benefits of the Semantic Web and to learn how to interact with semantically enriched knowledge.

Conceptually, Magpie research emphasized the importance of designing user-centered tools that fulfill user needs without introducing high overheads in terms of usability or processing delay. The Magpie application supporting the students of climatology has showed how Semantic Web research can contribute to the challenges facing other disciplines—in this case, e-learning and open distance education. Indeed more demonstrators like Magpie are needed to showcase the numerous benefits of the Semantic Web technologies and languages in people's everyday activities.

The second lesson we learned relates to deploying Magpie applications in the real world for real users. Like any other knowledge-based system, which attempts to deal with the open world of the Web, rather than with a relatively narrow context of a closed, constrained world, the biggest challenge for Magpie has been to take into account and address the evolution (or maintenance) of knowledge.

Knowledge maintenance has emerged in our research from the need to make ontology- and lexicon-based annotation of instances in a web page less brittle. In order to improve the performance of knowledge maintenance in a Magpie-based application, we leveraged Magpie's basic capability of plugging-in third-party semantic tools as specialized services available for a particular application. Simply by choreographing together the specialized engines we achieved different aspects of knowledge evolution depending on the degree of service integration.

Knowledge that has been recognized as relevant can be (i) *transient*—only displayed in a user's dedicated display, (ii) *persistent*—displayed and stored for an individual user, or (iii) *group-shared*—the shared ontological commitments are updated and validated. This variability is useful especially for managing and evolving different types of knowledge in a specific organization, because it takes into account not only knowledge-level processes but also social nature of knowledge evolution.

The third lesson learned then elaborates on the variability of processes that can be grouped under the umbrella of knowledge maintenance. The simplest step towards evolving knowledge is to deploy IE techniques and acquire new instances or concepts for a particular ontology. This step has been demonstrated using the C-PANKOW IE engine. More complex step towards evolving knowledge in a particular KB is to complement IE with the validation of knowledge chunks. Again, there are many ways in which this type of quality assurance may be implemented. We have experimented with a modular approach. This consisted of linking C-PANKOW to another IE tool (Armadillo) that used the principle of information redundancy on the Web to validate proposed knowledge chunks and to extend knowledge about these proposals.

A new challenge arising from integrating two or more originally independent tools into a semantic application relates to constructing meaningful data flow between the tools and the end user. In our experiments, the flow was designed manually with C-PANKOW and Armadillo interacting through a joint data repository. Pragmatically, it would be desirable to have such data flows constructed dynamically. This challenge, however, is already tackled by the research into Semantic Web services and their aggregation, integration of choreography (see for instance, IRS-III research [8]). These aspects although highly relevant to developing integrated semantic applications, are not the primary focus of our research here. Hence, they were not discussed in this article in any detail.

Another approach to maintaining knowledge may involve dedicated algorithms that are capable of resolving co-references within a particular KB, and thus add relations among different aspects related to the same (co-referenced) entity. Yet another approach to supporting the evolution of knowledge has simultaneously emerged in IBM's UIMA project [25]. In the case of UIMA, the authors decided to integrate multiple processes related to knowledge maintenance into one integrated architecture/engine. Both approaches are valuable. The modular one we piloted might be more flexible and extensible; the tightly integrated IBM's strategy may be more powerful and efficient. It is likely that the choice of strategy for evolving knowledge would depend on the context in which maintenance occurs. For organization intranets tight integration may be preferred; for extranets and the Web modularity might be a more robust way to proceed.

A potentially interesting lesson comes from the debate on choosing modular and specialized services as opposed to tightly integrated approaches. Namely, the modular approach allows some of the services involved in evolving knowledge to use statistical techniques, where other services may rely more on social trust. It seems that knowledge evolution would need to be investigated from two complementary perspectives—(i) formal knowledge evolution assuring consistency in a KB, and (ii) social knowledge evolution assuring the validity of knowledge in a KB.

Attention as opposed to information is now widely acknowledged to be the scarce resource in the Internet age. Consequently, tools that can leverage semantic resources to take some of the burden of the interpretation task from the human user are going to be of great use. Tools that are able to seamlessly bridge the "traditional" Web of documents and the various aspects of the Semantic Web are still in scarce supply. Such tools, however, are critical for several reasons. First, they bootstrap the Semantic Web. Second, they provide what McDowell et al. call instant gratification to the end users [39], thus motivating them to contribute to "semanticizing" the Web by annotating. We believe that Magpie is a step towards achieving the vision and acts as a bridge between two evolutionary stages of technologies for interacting with data, information and knowledge in an open, distributed world.

A lesson has been also learned on the level of tools facilitating knowledge maintenance. While there are many techniques and strategies for knowledge discovery, representation and reuse, the maintenance of knowledge is in its infancy. We are not aware of any major research into robust and scalable knowledge maintenance. The existing approaches using KB merging and versioning techniques are helpful, but they only address a small part of the knowledge processing chain.

As we have shown in this paper, the open architecture of the Magpie framework is not constrained to a single interactive channel, modality or type of knowledge. On the contrary, one framework is able to support and sustain modalities such as on-demand and "push". The Magpie framework is also capable of catering for diverse activities such as information or document retrieval, rule- or data patter-driven reasoning, and various degrees of knowledge maintenance. New techniques for IE, text analysis, knowledge validation or relationship discovery will surely emerge in the near future. Magpie's services framework facilitates the low-cost upgrade to these future technologies without any major re-design of the existing user components.

8. Projections to the future

The Semantic Web is gaining momentum and more semantic data is available online. This has an impact on the application development strategies. Magpie originates in the era before the aforementioned momentum became visible, so its assumption of no or little semantic mark-up available is now obsolete. The momentum implies that the new generation of Semantic Web application needs to work with more heterogeneous and distributed semantic data. Hence, another design principle (a single ontology) is challenged by this environment consisting of distributed and networked ontologies. Thus, the new generation of Magpie currently developed aims to overcome the "single ontology" limitation by exploiting the wealth of semantic data available online. The goal is to dynamically find and combine online semantics that are relevant to the current task performed by the specific application. This would allow cross-domain and extended coverage of semantic browsing, for instance.

The idea of exploiting the Web (and the Semantic Web) as a large source of background knowledge was prototypically tested (see Section 4.2) by us, but has also appeared in several recent works concerning generic tasks (e.g. sense disambiguation or ontology matching). For example, Alani proposed a method for ontology learning that relies on reusing ontology modules from online ontologies relevant to keywords from a user query [2]. Similarly, the use of the open Web as an online source of background knowledge for ontology mapping is reported in [45].

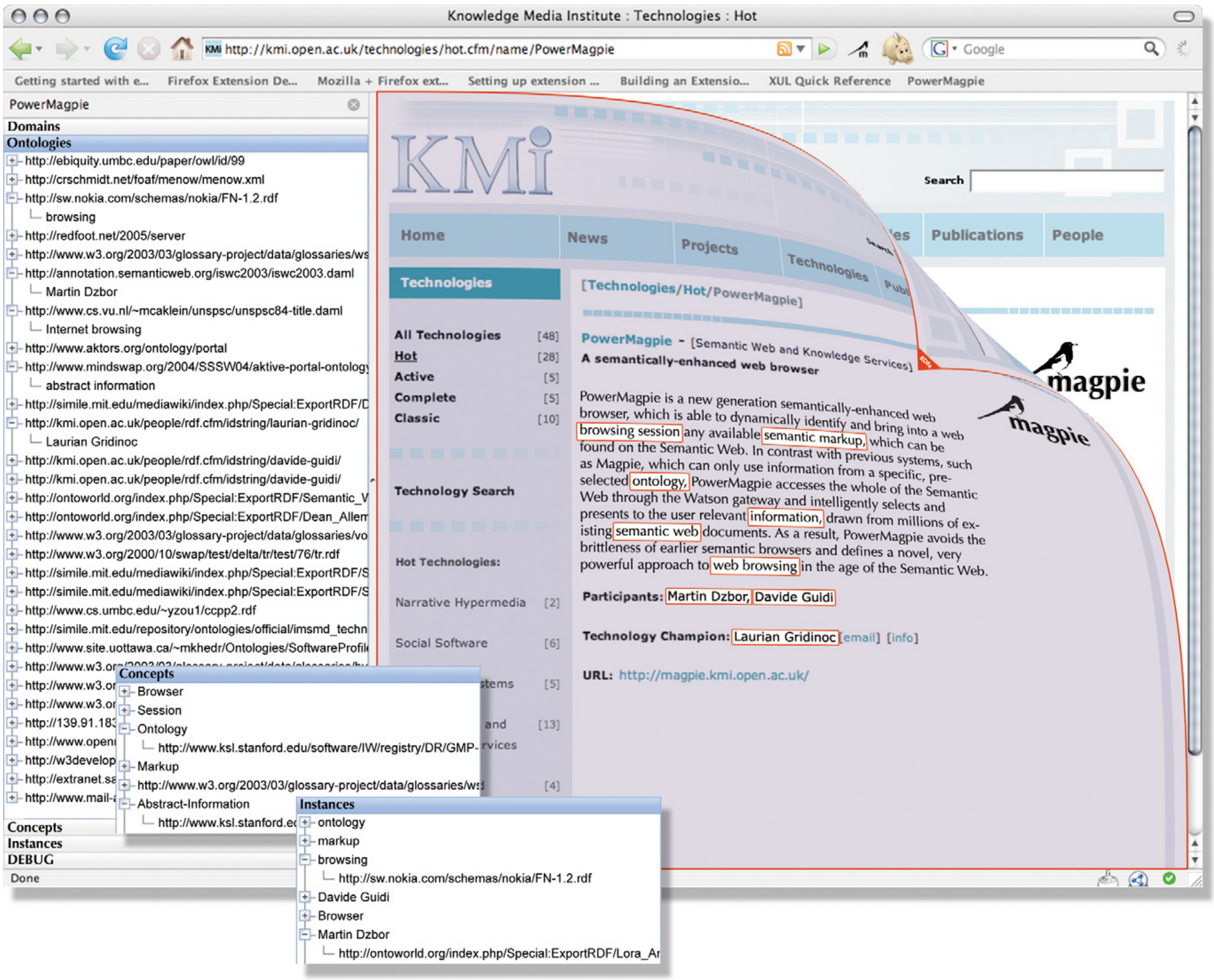
In the real world, it is unlikely that anybody would use only one ontological perspective at any one time. Translated to Magpie and any other semantic browsing tools, this means bringing in the capability to work with multiple ontologies simultaneously. Not only that; in the open Semantic Web, it is unlikely that all ontologies and various lexicons derived from those ontologies would reside at the same location. Ontological resources are geographically dispersed, networked and richly interlinked. Given this, it is no longer sufficient for the user to *choose* ontology. Users may want to *create their individual viewpoint* from many networked ontological components. They may want to do it on the fly, rather than relying on knowledge engineers.

Furthermore, one needs to combine semantic mark-up available within the accessed resources with the external semantic assertions coming, e.g. from third-party ontologies. A collateral effect of the emerging user-centered tools and increasingly pragmatic Semantic Web applications is that our initial premise of little a priori semantic mark-up available is no longer valid to the extent that it was in early 2000s. However, tools like Magpie may benefit from this—the existing mark-up (e.g. created by the document authors) can be maintained/evolved using the automatically discovered annotations. A new challenge would arise from the need to reconcile the differences and to combine these multiple sources in a manner that is transparent and useful for the end user.

Moreover, the idea tested by us of interlinking semantic annotation, semantic browsing and semantic services is gaining popularity. Annotation is no longer a separate objective in its own right; most of the new annotation tools aim to offer additional services; e.g. validation or consistency checking. A major challenge stems from the need to make the association between semantic services and semantic mark-up more open, flexible and simpler. Clearly, more usable techniques are needed for marking up, discovering and deploying services, so that they can be used in a new generation of open semantic applications.

One particular area, where we see our research on Semantic Web application development framework going, is toward broadening and further loosening the interfaces between the individual components and techniques. So rather than linking techniques in an ad hoc way we presume the next step would be to form an *infrastructure* of components, techniques and data sets, on top of which specific applications (such as Semantic Web browsers or question answering tools) could be built. Hence, one direction where previous research on Magpie is heading involves the establishment of a *Semantic Web gateway*.

The key point is to build an infrastructure that enables seamless and transparent access to content with little additional costs to the specialized, user-facing semantic applications. Two examples of technologies contributing to building of such an infrastructure are the Watson—Semantic Web Gateway [14]



PowerMagpie

Domains

Ontologies

- http://ebiquity.umbc.edu/paper/owl/id/99
- http://crschmidt.net/foaf/menow/menow.xml
- http://sw.nokia.com/schemas/nokia/FN-1.2.rdf
  - browsing
- http://redfoot.net/2005/server
- http://www.w3.org/2003/03/glossary-project/data/glossaries/ws
- http://annotation.semanticweb.org/iswc2003/iswc2003.daml
  - Martin Dzbor
- http://www.cs.vu.nl/~mcaklein/unspsc/unspsc84-title.daml
  - Internet browsing
- http://www.aktors.org/ontology/portal
- http://www.mindswap.org/2004/SSSW04/aktive-portal-ontology
  - abstract information
- http://simile.mit.edu/mediawiki/index.php/Special:ExportRDF/D
- http://kmi.open.ac.uk/people/rdf.cfm/idstring/laurian-gridinoc/
  - Laurian Gridinoc
- http://kmi.open.ac.uk/people/rdf.cfm/idstring/davide-guidi/
- http://kmi.open.ac.uk/people/rdf.cfm/idstring/davide-guidi/
- http://ontoworld.org/index.php/Special:ExportRDF/Semantic_V
- http://ontoworld.org/index.php/Special:ExportRDF/Dean_Allem
- http://www.w3.org/2003/03/glossary-project/data/glossaries/vo
- http://www.w3.org/2000/10/swap/test/delta/tr/test/76/tr.rdf
- http://simile.mit.edu/mediawiki/index.php/Special:ExportRDF/S
- http://simile.mit.edu/mediawiki/index.php/Special:ExportRDF/S
- http://www.cs.umbc.edu/~yzou1/ccpp2.rdf
- http://simile.mit.edu/repository/ontologies/official/imsmd_techn
- http://www.site.uottawa.ca/~mkhedr/Ontologies/SoftwareProfile
- http://www.w3.or
- http://www.w3.or
- http://www.w3.or
- http://139.91.183
- http://www.openi
- http://w3develop
- http://extranet.sa
- http://www.mail-a

Concepts

Instances

DEBUG

Concepts

- Browser
- Session
- Ontology
  - http://www.ksl.stanford.edu/software/IW/registry/DR/GMP-
- Markup
- http://www.w3.org/2003/03/glossary-project/data/glossaries/ws
- Abstract-Information
  - http://www.ksl.stanford.ed

Instances

- ontology
- markup
- browsing
  - http://sw.nokia.com/schemas/nokia/FN-1.2.rdf
- Davide Guidi
- Browser
- Martin Dzbor
  - http://ontoworld.org/index.php/Special:ExportRDF/Lora_Ar

Technologies

All Technologies	[48]
Hot	[28]
Active	[5]
Complete	[5]
Classic	[10]

Technology Search

Hot Technologies:

Narrative Hypermedia	[2]
Social Software	[6]
stems	[5]
and rvices	[13]
	[4]

[Technologies/Hot/PowerMagpie]

PowerMagpie - [Semantic Web and Knowledge Services]

A semantically-enhanced web browser

PowerMagpie is a new generation semantically-enhanced web browser, which is able to dynamically identify and bring into a web browsing session any available semantic markup, which can be found on the Semantic Web. In contrast with previous systems, such as Magpie, which can only use information from a specific, pre-selected ontology, PowerMagpie accesses the whole of the Semantic Web through the Watson gateway and intelligently selects and presents to the user relevant information, drawn from millions of existing semantic web documents. As a result, PowerMagpie avoids the brittleness of earlier semantic browsers and defines a novel, very powerful approach to web browsing in the age of the Semantic Web.

Participants: Martin Dzbor, Davide Guidi

Technology Champion: Laurian Gridinoc [email] [info]

URL: http://magpie.kmi.open.ac.uk/

Fig. 7. A snapshot of PowerMagpie embodying some of the visions discussed in this article.

combined with our prototypic PowerMagpie tool.[4] The combination of these two tools enables us to make a progress on the capability to apply multiple ontological perspectives in multiple user contexts, which was one of our criteria in Section 3.

The main purpose of Watson is to enable advanced Semantic Web applications to access and use ontologies that may be distributed throughout the Web. Another purpose of Watson is to collect in-depth knowledge about the distributed semantic content available for the purposes of browsing, question answering, etc. Watson's contribution to satisfying our criteria from Section 3.2 is in discovering ontologies and other semantic content by means of commonly used data harvesting techniques. These may produce a large quantity of input, so the actual added value of this envisaged framework is the semantic and qualitative analysis of the harvested content.

PowerMagpie (Fig. 7), our prototype semantic browser to succeed Magpie relies on the generic Watson infrastructure. The main task for Watson in this semantic "power browsing" tool is to identify and make available ontologies that are interesting or otherwise relevant to the content of the web page currently visited by the user. This reliance is due to the fact that we want to make the semantic layering step introduced for Magpie automated, and for that reason, we need to distinguish the ontologies of different quality, topic coverage, expressivity, etc., rather than merely finding any semantic content containing a given keyword.

Once relevant ontologies are filtered, their content is then presented in the PowerMagpie's end user interface in a variety of flavours; e.g. one can apply one ontology to create a semantic layer, or one can take a theme- or instance-driven approach to creating semantic layers with different purposes. The layers become more dynamic and resemble skins of an application rather than annotations, as shown in a snapshot from the early prototype in Fig. 7.

As Fig. 7 shows, the requirement to parametrize an application with multiple ontologies is indeed realistic, and is likely to increase the degree of flexibility in interacting with semantic content using the web browsing metaphors. PowerMagpie is, thus, an early step towards the visions we presented in this

[4] More details on the PowerMagpie technology are on http://magpie.kmi.open.ac.uk.

reflective paper on the previous version of the Magpie technology.

Acknowledgments

The Magpie research has been supported by the climate*prediction*.net, Dot.Kom, Knowledge Web and Advanced Knowledge Technologies (AKT) projects. Climate*prediction*.net was sponsored by the UK Natural Environment Research Council and UK Department of Trade e-Science Initiative. Dot.Kom (Designing Adaptive Information Extraction from Text for Knowledge Management) by the IST Framework 5 grant no. IST-2001-34038. Knowledge Web is an IST Framework 6 Network of Excellence (grant no. FP6-507482). AKT is an Interdisciplinary Research Collaboration (IRC) sponsored by the UK Engineering and Physical Sciences Research Council by grant no. GR/N15764/01.

References

[1] S. Agarwal, S. Handshuh, S. Staab, Surfing the service web, in: Proceedings of the 2nd International Semantic Web Conference, Florida, USA, 2003.
[2] H. Alani, Ontology construction from online ontologies, in: Proceedings of the 15th International WWW Conference, Scotland, UK, 2006.
[3] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Sci. Am. 279 (5) (2001) 34–43.
[4] T. Bray, J. Paoli, C.M. Sperberg-McQueen, et al., Extensible Markup Language (XML) 1.0, second ed., 2000 (cited September 2002).
[5] D. Brickley, R. Guha, RDF Vocabulary Description Language 1.0: RDF Schema, World Wide Web Consortium, 2004, URL: http://www.w3.org/TR/rdf-schema.
[6] D.C. Brown, I'm scruffy and at the knowledge level, Int. J. Design Comput. 1 (1) (1998), p. column.
[7] P. Buitelaar, T. Eigner, Semantic navigation with VIeWs, in: UserSWeb: Workshop on User Aspects of the Semantic Web, Crete, 2005.
[8] L. Cabral, J. Domingue, S. Galizia, et al., IRS-III: a broker for semantic web services based applications, in: Proceedings of the 5th International Semantic Web Conference, Georgia, USA, 2006.
[9] L. Carr, S. Bechhofer, C. Goble, et al., Conceptual linking: ontology-based open hypermedia, in: Proceedings of the 10th International WWW Conference, Hong-Kong, 2001.
[10] P. Cimiano, S. Handshuh, S. Staab, Towards the self-annotating web, in: Proceedings of the 13th International WWW Conference, ACM Press, New York, 2004.
[11] P. Cimiano, G. Ladwig, S. Staab, Gimme' the context: context-driven automatic semantic annotation with C-PANKOW, in: Proceedings of the 14th International WWW Conference, Japan, 2005.
[12] F. Ciravegna, A. Dingli, D. Guthrie, et al., Integrating information to bootstrap information extraction from web sites, in: IJCAI Workshop on Information Integration on the Web, Mexico, 2003.
[13] H. Cunningham, D. Maynard, K. Bontcheva, et al., GATE: a framework and graphical development environment for robust NLP tools and applications, in: 40th Anniversary Meeting of the Association for Computational Linguistics (ACL), Pennsylvania, USA, 2002.
[14] M. d'Aquin, M. Sabou, M. Dzbor, et al., WATSON: a gateway for the semantic web, in: Posters of the 4th European Semantic Web Conference, Austria, 2007.
[15] S. Dill, N. Eiron, D. Gibson, et al., SemTag and Seeker: bootstrapping the semantic web via automated semantic annotation, in: Proceedings of the 12th International WWW Conference, ACM Press, Hungary, 2003.
[16] L. Ding, T. Finin, Characterizing the Semantic Web on the Web, in: Proceedings of the 5th International Semantic Web Conference, Springer Verlag, Georgia, USA, 2006.
[17] J. Domingue, M. Dzbor, E. Motta, Semantic layering with Magpie, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies in Information Systems, Springer Verlag, 2003.
[18] J. Domingue, M. Dzbor, E. Motta, Collaborative Semantic Web browsing with Magpie, in: Proceedings of the 1st European Semantic Web Symposium, Greece, 2004.
[19] J. Domingue, M. Dzbor, E. Motta, Magpie: supporting browsing and navigation on the Semantic Web, in: Proceedings of the 13th Conference on Intelligent User Interfaces (IUI), Madeira, Portugal, 2004.
[20] M. Dzbor, J. Domingue, E. Motta, Magpie: towards a Semantic Web browser, in: Proceedings of the 2nd International Semantic Web Conference, Florida, USA, 2003.
[21] M. Dzbor, E. Motta, J. Domingue, Opening up Magpie via semantic services, in: Proceedings of the 3rd International Semantic Web Conference, Japan, 2004.
[22] O. Etzioni, M. Cafarella, D. Downey, et al., Methods for domain-independent information extraction from the web: an experimental comparison, in: Proceedings of the 19th AAAI Conference, California, USA, 2004.
[23] J. Euzenat, P. Shvaiko, Ontology Matching, Springer Verlag, Heidelberg, Germany, 2007, 333 pp.
[24] D. Fensel, E. Motta, Structured development of problem solving methods, IEEE Trans. Knowledge Data Eng. 13 (6) (2001) 913–932.
[25] D. Ferrucci, A. Lally, Building an example application with the Unstructured Information Management Architecture, IBM Syst. J. 43 (3) (2004) 455–475.
[26] J. Gennari, M.A. Musen, R. Fergerson, et al., The evolution of Protege-2000: an environment for knowledge-based systems development, Int. J. Hum.-Comp. Studies 58 (1) (2003) 89–123.
[27] T.R. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition 5 (2) (1993) 199–221.
[28] J. Grudin, Groupware and social dynamics: eight challenges for developers, Commun. ACM 37 (1) (1994) 92–105.
[29] S. Handschuh, S. Staab, A. Maedche, CREAM—creating relational metadata with a component-based, ontology driven annotation framework, in: International Semantic Web Working Symposium (SWWS), California, USA, 2001.
[30] F. Hayes-Roth, D.A. Waterman, D.B. Lenat (Eds.), Building Expert Systems, Addison-Wesley, Massachusetts, USA, 1983.
[31] A. Hogue, D.R. Karger, Thresher: automating the unwrapping of semantic content from the World Wide Web, in: Proceedings of the 14th International WWW Conference, ACM Press, Japan, 2005.
[32] D. Huynh, S. Mazzocchi, D.R. Karger, Piggy Bank: experience the Semantic Web inside your web browser, in: Proceedings of the 4th International Semantic Web Conference, Ireland, 2005.
[33] J. Kahan, M.-R. Koivunen, E. Prud'Hommeaux, et al., Annotea: an open RDF infrastructure for shared web annotations, in: Proceedings of the 10th International WWW Conference, Hong-Kong, 2001.
[34] Y. Kalfoglou, H. Alani, M. Schorlemmer, et al., On the emergent Semantic Web and overlooked issues, in: Proceedings of the 3rd International Semantic Web Conference, Japan, 2004.
[35] A. Kay, User interface: a personal view, in: B. Laurel (Ed.), The Art of Human–Computer Interface Design, Addison-Wesley, Massachussets, 1990, pp. 191–207.
[36] V. Lanfranchi, F. Ciravegna, D. Petrelli, Semantic Web-based document: editing and browsing in AktiveDoc, in: Proceedings of the 2nd European Semantic Web Conference, Greece, 2005.
[37] H. Lieberman, C. Fry, L. Weitzman, Exploring the web with reconnaissance agents, Comm. ACM 44 (8) (2001) 69–75.
[38] P.J. Lynch, Visual design for the user interface (Part 1): design fundamentals, J. Biocommun. 22 (1) (1994) 22–30.
[39] L. McDowell, O. Etzioni, S.D. Gribble, et al., Mangrove: enticing ordinary people onto the Semantic Web via instant gratification, in: Proceedings of the 2nd International Semantic Web Conference, Florida, USA, 2003.
[40] E. Motta, J. Domingue, L. Cabral, et al., IRS-II: a framework and infrastructure for Semantic Web services, in: Proceedings of the 2nd International Semantic Web Conference, Florida, USA, 2003.

[41] D. Norman, The Psychology of Everyday Things, Basic Books, New York, USA, 1988.

[42] B. Popov, A. Kiryakov, A. Kirilov, et al., KIM—semantic annotation platform, in: Proceedings of the 2nd International Semantic Web Conference, Florida, USA, 2003.

[43] D. Quan, D. Huynh, D.R. Karger, Haystack: a platform for authoring end user Semantic Web applications, in: Proceedings of the 2nd International Semantic Web Conference, Florida, USA, 2003.

[44] D. Quan, D.R. Karger, How to make a semantic web browser, in: Proceedings of the 13th International Conference on World Wide Web, ACM Press, New York, USA, 2004.

[45] R.M. Sabou, M. d'Aquin, E. Motta, Using the Semantic Web as background knowledge for ontology mapping, in: Proceedings of the Ontology Mapping Workshop (collocated with ISWC 2006), Georgia, USA, 2006.

[46] N.R. Shadbolt, AKT: A Manifesto of an EPSRC Interdisciplinary Research Collaboration, 2000.

[47] A. Sheth, C. Bertram, D. Avant, et al., Managing semantic content for the web, IEEE Internet Comput. 6 (4) (2002) 80–87.

[48] H. Takeda, I. Ohmukai, Building semantic web applications as information/knowledge sharing systems, in: UserSWeb: Workshop on User Aspects of the Semantic Web, Crete, Greece, 2005.

[49] V.S. Uren, P. Cimiano, E. Motta, et al., Browsing for information by highlighting automatically generated annotations: user study and evaluation, in: Proceedings of the 3rd Knowledge Capture Conference, Canada, 2005.

[50] F. van Harmelen, J. Hendler, I. Horrocks, et al., OWL Web Ontology Language Reference, 2002 (cited March 2003).