Din din! The (Semantic) Turkey is served!

Maria Teresa Pazienza, Noemi Scarpato, Armando Stellato, Andrea Turbati

ART Group, Dept. of Computer Science, Systems and Production
University of Rome, Tor Vergata
Via del Politecnico 1, 00133 Rome, Italy
{pazienza, scarpato, stellato, turbati}@info.uniroma2.it

Abstract. From its first introduction in this same conference, the original prototype of the Semantic Bookmarking tool Semantic Turkey has undergone a deep and extensive revision process, breaking the boundaries of its original intents and going more and more towards an extensible platform for Knowledge Management and Acquisition based on Semantic Web technologies. Following its recent official release, we discuss here the main innovations of this system, its potential applications and future plans for its improvement

1. Introduction

The Semantic Web is becoming aver and ever a concrete reality: with SPARQL reaching W3C recommendation early this year [14], languages for data representation and querying have finally reached standardization, while interests and research in Semantic Web technologies have definitely migrated from mere ontology development (which has now met industry standards) aspects to the discovery and devise of applications which can both show and exploit Semantic Web full potential.

With this scenario in mind, we have worked towards the definition of a Semantic Web browser extension which is two-fold in its offer: by first, being of interest for ontology developers and domain experts, since it aims at facilitating the process of knowledge acquisition and development, and, on the other side, providing an extensible infrastructure over which SW applications, needing and relying on rock-solid web browsing functionalities as well as on RDF management capacities, can be developed and deployed.

These objectives have been pursued during a two-years work of finalization and reengineering of Semantic Turkey [9]: a Semantic Web Browser extension which had previously been introduced – in its first demonstrating prototype – inside this same conference [8].

In this work, we discuss the main innovations which accompanied the official release of Semantic Turkey, show its potential applications also by referencing our experience in different research collaborations, and present future plans for its improvement. The next section contains a very general introduction to the characteristics of the Semantic Turkey framework; for a thorough review of related works and for a detailed description of the rationales behind Semantic Turkey, we suggest [8] as an introductory reading.

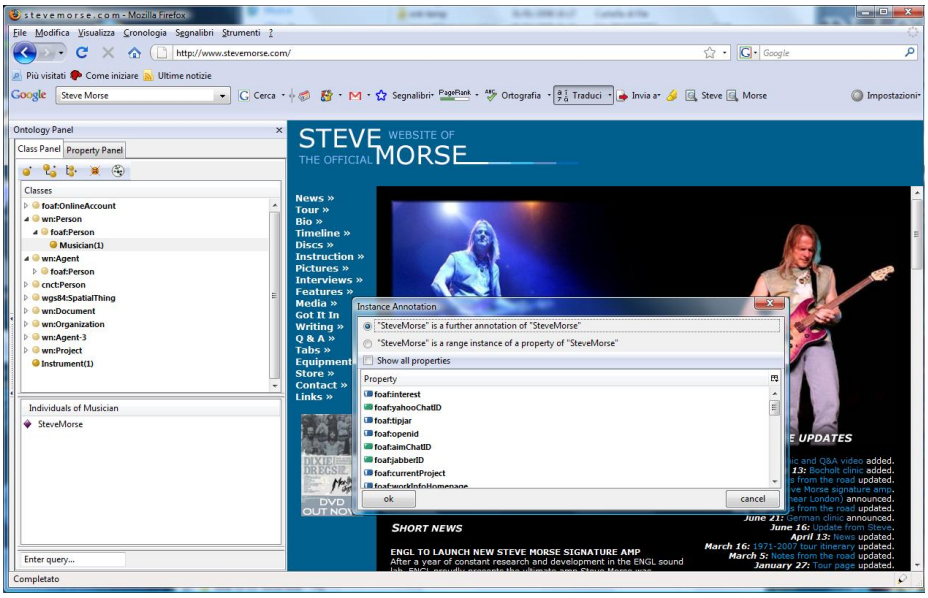

Fig. 1 Semantic Bookmarking with Semantic Turkey

2. From Semantic Bookmarking to Knowledge Management and Acquisition

Semantic Turkey was born inside a national project – funded by the FILAS agency (Finanziaria Laziale di Sviluppo) under contract C5748-2005 – focused on innovative solutions for browsing the web and for collecting and organizing the information observed during navigation.

The prototype for the project immediately took the form of a Web Browser extension allowing users to annotate information from visited web sites and organize it according to a personally defined domain model: Semantic Turkey paradigmatic innovation was in fact to "obtain a clear separation between (acquired) knowledge data (the WHAT) and web links (the WHERE)" pointing to it. That is, to be able, through very easy-to-use drag'n'drop gestures, to *select* textual information from web pages, *create* objects in a given domain and *annotate* their presence in the web by keeping track of the selected text and of its provenience (web page *url*, *title* etc…). We coined the expression "semantic bookmarking" for this kind of activity.

Due to its proverbial extendibility, the Firefox platform (http://www.mozilla.com/en-US/firefox/) had been chosen as the hosting browser for our application, while Semantic Web standards and technologies were the natural candidate for representing its knowledge model.

Semantic Turkey (Fig. 1) was thus born. Standing on top of mature results from research on Semantic Web technologies, like Sesame [2] and OWLim [12] as well as on a robust platform such as the Firefox web browser, ST (Semantic Turkey)

differentiates from other existing approaches which are more specifically tailored respectively towards knowledge management and editing [7], semantic mashup and browsing [5, 10] and pure semantic annotation [3, 11], by introducing a new dimension which is unique to the process of building new knowledge while exploring the web to acquire it.

By focusing on this aspect, which has been further investigated in the two years of finalization leading to the current release, we went beyond the original concept of Semantic Bookmarking and tried to amplify the potential of a new Knowledge Management and Acquisition System: we thus aimed at reducing the impedance mismatch between domain experts and knowledge investigators on the one side, and knowledge engineers on the other, providing them with a unifying platform for acquiring, building up, reorganizing and refining knowledge.

3. Usability

The first Semantic Turkey prototype was a conceptual bookmarking system which based its backing data on the OWL standard, though subject to an highly constrained model: there were only one ontology which could be edited by the user, and it was possible to specify only *unfaceted* object properties for relating objects of the domain; ontology editing was limited to deletion and renaming of individuals added as semantic bookmarks.

The final project moved to an open editor for data modeled upon languages of the RDF family, allowing the exploitation of almost all of those language potentialities (currently, it lacks of anonymous resources and OWL descriptors producing anonymous classes). To allow maximum flexibility, every element in the ontology can now be added through the advanced bookmarking/annotation functionalities (see Fig. 2) or directly through the ontology editor (in both cases, further annotations can be added later to the created objects).

Fig. 2 shows the different annotation/knowledge acquisition possibilities offered by the functionalities based on interaction with the hosting web browser. In the new version of ST, support for all kind of properties has been introduced and reflected in the bookmarking facility: when a portion of text is selected from the page and dragged over an individual, the user may choose (as in the old version) to add a new annotation for the same individual or to use the annotation to fill one property slot for it. In the second case, the user can now choose from a list of properties (see small window in Fig. 1) the one which will be filled: this list includes those properties having their rdfs:domain including one of the types of the selected instance, but may be extended to cover all properties (letting the inference engine do the rest). If the property selected for enrichment is an object property, the user is prompted with a class tree (rooted on the rdfs:range of the selected property) and is given the possibility of creating a new individual named after the text selected for the annotation or to choose an existing one: in both cases the selected individual is bound – through the chosen property – to the one where he originally dropped the text; a bookmark is also added for it, pointing to the page where the object has been observed. Even in this case, the user may choose to visualize the entire class tree and

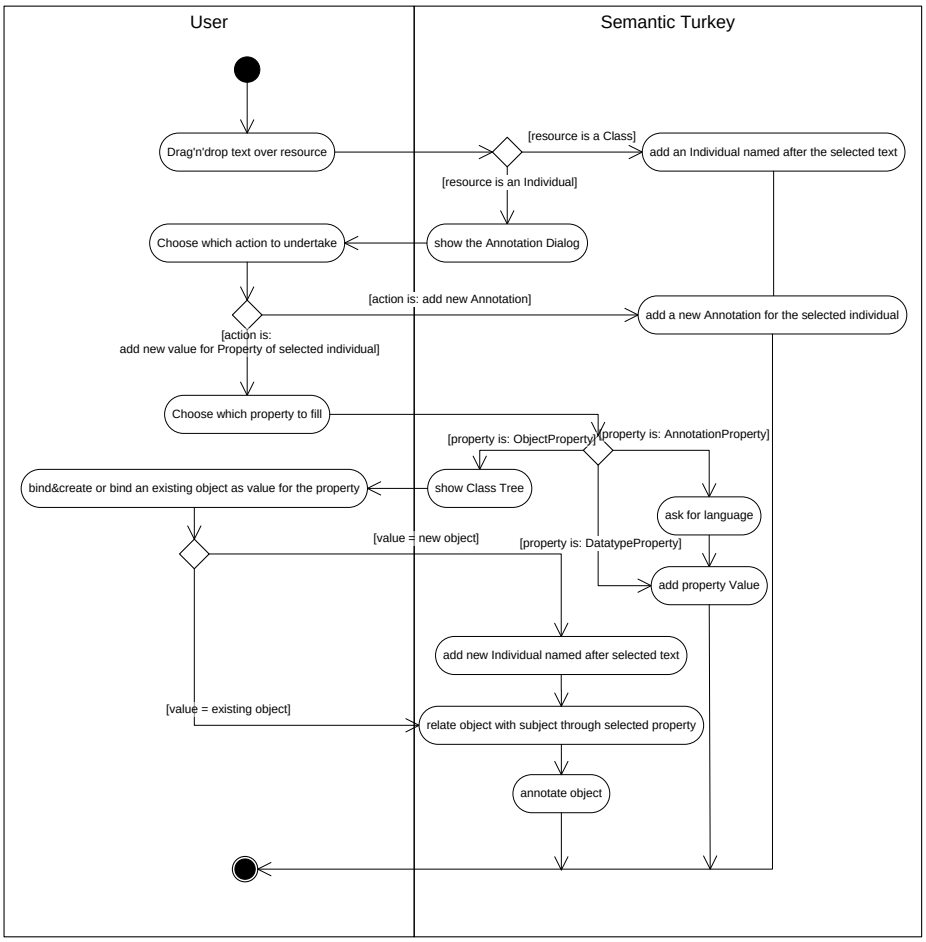

Fig. 2 Activity diagram for semantic bookmarking/annotation

not the one dominated by the range of the property: the inference engine will automatically assign the pointed instance to that range.

4. Knowledge Model

The prototype offered three architectural layers, consisting in: an *application layer*, containing the annotation ontology, describing the concepts for storage and handling of semantic annotations (this layer was hidden from the user), the *top layer*, providing read-only ontological descriptors to be used inside a shared context (it was expressly thought inside our work for FILAS, to be adopted by different users sharing information among them), and the *user layer*, allowing users to customize their ontology and to add instance data.

Out of the specific context where the prototype has been developed, we abandoned the *top layer*, in favor of the traditional (as of any ontology editor) imported/working ontology partitioning, where the latter as write permission for the user. We thus added support for importing ontologies from web/local files and included support for management of a local mirror where it is possible to store and retrieve ontologies.

The *Application layer* now beneficiates of the support for extendibility (see the following section on Architecture), so that extensions of Semantic Turkey can declare themselves to be based on new application ontologies, so that these will be added to the application layer and be treated accordingly before the whole ontology model is loaded

5. Architecture

While the technologies adopted for the realization of Semantic Turkey are mainly the same of its original prototype, its architecture has evolved since then. As shown in Fig. 3, all the main modifications have been introduced with the ultimate goal of supporting platform extendibility.

The whole extension mechanism is obtained by a proper combination of the Mozilla extension framework (which is used to extend the user interface, drive user interaction and add/modify browser functionalities of ST) and the OSGi java extension framework [13] (providing extensions capabilities for the service and data layers of the architecture). OSGi compliance is obtained through the OSGi implementation developed inside the Apache Software Foundation, called Felix (felix.apache.org/).

Two main extension points have been introduced: a *Service extension* and a *Repository Extension* (dotted flat boxes in the architecture). The first one allows for the development of arbitrary services which can be added dynamically to the system. Extensions of this type typically need to realize both a client extension through Mozilla technology, by adding new functionalities (and hooks for them in the user interface) to the system, as well the corresponding Service which is added dynamically through OSGi.

The second kind of extension provides openness to different triples store technologies; Semantic Turkey is in fact no more strictly based on the Sesame + OWLim libraries for RDF management, but features proprietary APIs for querying the managed ontologies. These API are defined through a set of interfaces, which can be implemented to adopt different triple stores. This can be of particular interest in specific scenarios where the target user has to connect to a specific triplestore, or where a service extension is being built by annexing an existing application, and in either case, these are based on a different triple store technologies.

Both kind of extensions are deployable as an xpi (cross-platform installers) packages which, once installed inside Firefox, are handled by Semantic Turkey extension discovery system, which extracts OSGi bundles and installs them in the main application. This assures easy installation for the user, which can install ST extensions as any other Firefox one, by dragging the xpi over Firefox and restarting the browser.

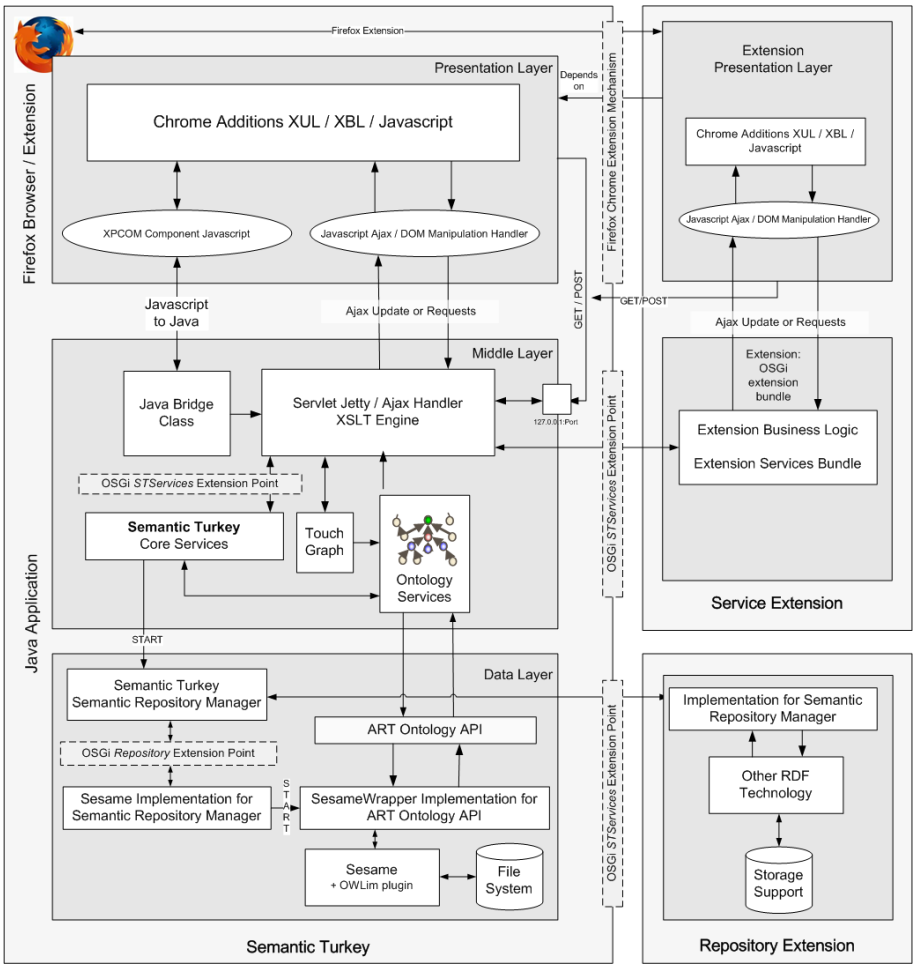

Fig. 3 Architecture of Semantic Turkey and of its extensions

6. Extensions

The combination of possibilities offered by the extension mechanisms of both Semantic Turkey and of its hosting web browser provides a flexible framework for rapid development of Web based Knowledge Management and Acquisition Systems.

Semantic Turkey has been used in different application domains, often introducing functionalities which, without support for extendibility, would have required heavy customization of the original tool. We report here a few projects and research collaborations which led to the development of an extension for Semantic Turkey:

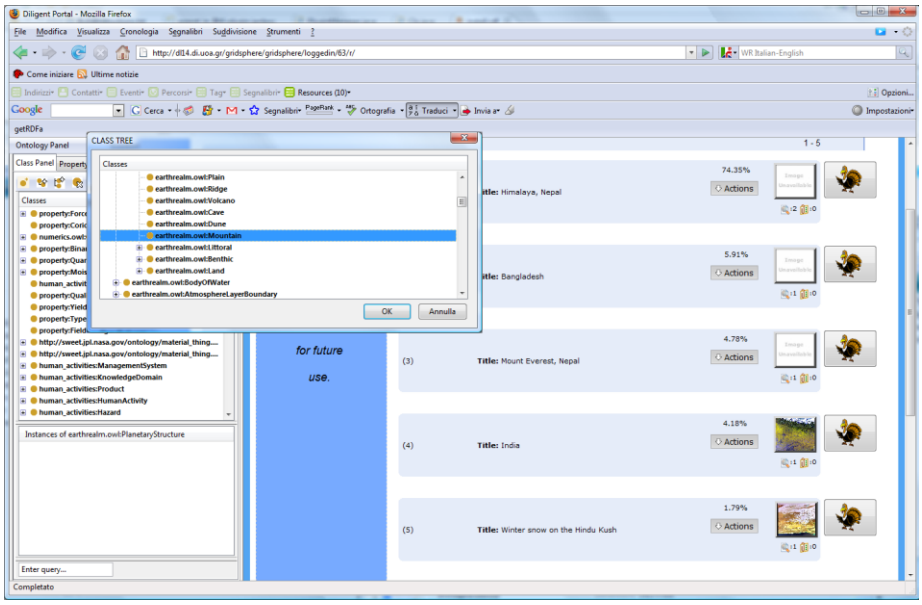

Fig. 4 recognizing, extracting and bookmarking RDFa data from the DILIGENT portal

6.1. EOAnnotator

Developed inside our collaboration with the European Space Agency (ESA), as subcontractors in the EU funded project Diligent (IST-004260), *EOAnnotator* [6] provides customized facilities for interaction with Portals for Earth Observation. This extension allows users to annotate information from the web according to the SWEET ontologies [15] developed by NASA, which can be used to propose new objects to be stored in the portal, as well as to retrieve data from the portal and store them in the personal ontology, thus obtaining a collection of "bookmarked objects". This last does not require manual annotation, since EOAnnotator is capable of recognizing RDFa code inside the Diligent EOPortal and to propose to the user which informative objects from the portal should be "bookmarked" inside his ontology. Following this work, a general purpose extension for managing and importing RDFa data and microformats (http://microformats.org/) is currently being developed.

6.2. RangeAnnotator

Semantic Turkey annotation mechanism produces *semantic bookmarks*, in that it keeps track of annotated pages, their association to ontology resources and of the *textual occurrences* of these resources in the page; this choice has been taken due to the high variability of web content, which made it useless to keep track of the exact position (which could vary in time) of strings inside web pages. Semantic Turkey in fact just finds back all the textual occurrences of objects which were annotated in a

given page every time that page is loaded, so if a page changes, it is easy to find them back, wherever these are positioned, unless they have been totally cancelled from the page.

The *RangeAnnotator* extension transforms Semantic Turkey into a true Semantic Annotation System, by replacing the standard annotation mechanism with one producing RangeAnnotations. The application ontology of Semantic Turkey already includes the concept of RangeAnnotation, which is defined as "an Annotation including range information" (that is: a *location* defined by two points, a start point and an end point). This range information can be implemented according to different formats and interpreted accordingly by a dedicated annotation extension. The current RangeAnnotator extension implements the RangeAnnotation concept by adopting Xpointers [4], thanks to the availability of a dedicated firefox library (http://xpointerlib.mozdev.org/) for handling this type of reference. RangeAnnotator can be used to produce *semantically annotated corpora of documents* and, under these circumstances, the selected collections of documents are expected to remain *unchanged*. Being developed outside of any specific context, RangeAnnotator is also the first extension which has been made publically available[1].

6.3. UIMA Web Annotator

This project, partially funded through an IBM UIMA Innovation Award (https://www-304.ibm.com/jct09002c/university/scholars/innovation/index.html) aims at integrating Semantic Turkey with the Unstructured Information Management Architecture (UIMA): a platform – originated at IBM and lately moved to an open source project incubating at the Apache Software Foundation (http://incubator.apache.org/uima) – for the creation, integration and deployment of unstructured information management solutions. The specific goal of the project is to transform Firefox+Semantic Turkey into a UIMA compliant CAS (Common Analysis Structure) annotator, so that users can produce annotated corpora of documents by annotating standard web pages, instead of textual surrogates as in the case of traditional UIMA CAS annotator. A CAS Editor is also included, and a utility for projecting data annotated with respect to an ontology towards a CAS is also present, based on a mapping which can be generated automatically and then refined by the user. *RangeAnnotator* extension is required to produce punctual RangeAnnotations, which can then be translated in the CAS according to different *range* formats.

6.4. Sayid (Semantic Annotation in Jurisprudence Domain)

This project, born inside a collaboration framework with CNIPA (Centro Nazionale per l'Informatica nella Pubblica Amministrazione), will lead to the development of a tool for annotating relations of *pertinence* between different laws (or part of laws), by accessing them from the web. Like the previous extension, *Sayid* requires the presence of *RangeAnnotator* to produce pointwise annotations.

[1] RangeAnnotator page can be accessed at the url: http://semanticturkey.uniroma2.it/extensions/rangeannotator

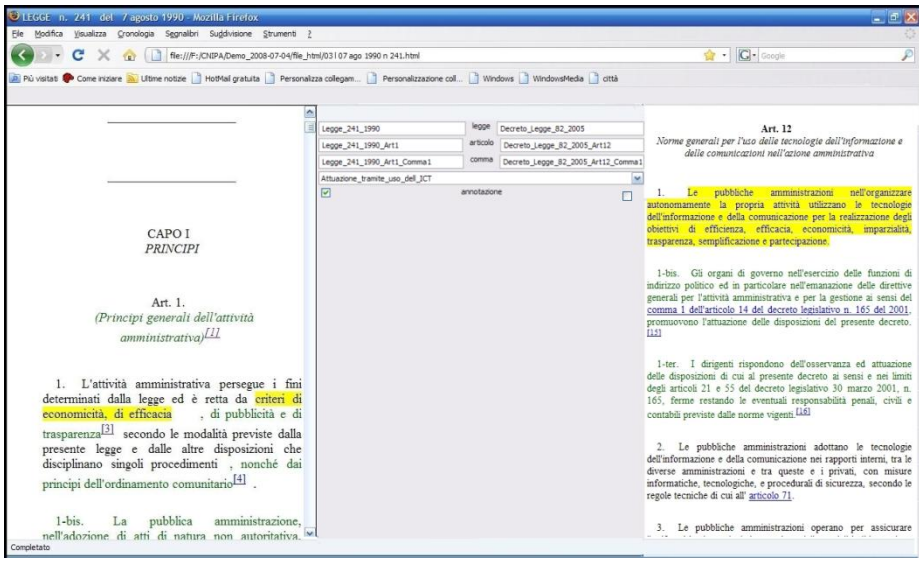

Fig. 5 relating portions of different laws with *Sayid* extension

This extension is an example of how Semantic Turkey core services can be exploited by a totally new application: *Sayid* is in fact a complete Firefox add-on relying on ST ontology management services more than an extension of this latter.

First of all, the Semantic Annotation mechanism is totally different from the standard one: in *Sayid* it is possible to bind laws through different relations of *pertinence*, but it is possible also to bind Annotations taken upon laws. *Annotations* become thus first class citizens in the domain ontology of this tool (while they are only implicitly accessible as web links in the standard ST).

Completely focused on this particular environment, the user interface of the tool is also totally original and does not extend the one offered by Semantic Turkey: it hides all the ontology editing capabilities of ST, statically adopts a specific ontology for handling concepts from jurisprudence and those needed for the annotation, and provides dedicated forms for managing them. Fig. 5 offers a screenshot taken after a user relating annotations from portions of two articles.

7. Discussion and Future Directions

Probably, the next step which research and development on this platform should take is to address the potentialities which have arisen by opening it up to full ontology development and to extendibility, and to further explore how these can combine at best with ST's inherent web access capabilities. On the other hand, both the above features are still in their infancy and we had to concentrate on guaranteeing robustness and consistency of their offer.

By considering ST no more as a Bookmarking tool, but rather as a Knowledge Acquisition platform, we cannot ignore important modeling axioms provided by the

OWL language (restrictions, set operators etc…which are currently not available for editing, though being properly processed at data&inference level) while support for SPARQL querying would be more than welcome.

The experiences that we have reported in the adoption of Semantic Turkey in different application scenarios have been a test bed for evaluating the real possibilities of extension development. The result is that, though far from perfect, the extension mechanism (together with the open service based approach) is flexible enough to allow for very different uses of the platform. For example, both the UIMA and Sayid extensions showed how it is possible to build completely new tools by working on the Firefox side, adding heavy weighted new services (e.g. access to the UIMA platform) and solely relying on the ontology management services provided by the platform. Furthermore, both of them showed potential increment of the platform by building extensions on top of over extensions (both of them rely on ST RangeAnnotator). In this sense, we understood the added value of an ontology development platform, comprehending high level data access and manipulation primitives which go beyond basilar RDF management provided by triple store libraries such as Jena or Sesame. Any of the actions performed by the user through the ontology editor is usually translated in several ontology editing primitives: for this reason, these high-level operations should not only be exposed as services, but provided instead as direct API for other extensions needing to rely on that level of abstraction.

Finally, the Range Annotator experience showed us the importance of going beyond basic service extendibility. At present time, by exploiting the Mozilla overlaying mechanism, RangeAnnotator *overrides* standard bookmarking requests from the client with calls to its specific annotation service. This results in:

- incompatibility with other similar extensions (both of them would try to override the client calls, with unpredictable results)
- no elegant switching solution (users currently need to deactivate Range Annotator if they want to revert back to standard bookmarking)
- duplication of code for describing common aspects of the annotation process: this aspect is correctly expressed in the annotation ontology, which features a generic Semantic Annotation concept which can be further specified, but provides no corresponding handles in the architecture

These issues point out the need for:

- replication of extension-point paradigm on the client
- dedicated extension points for further specification of existing functionalities (such as the one for taking annotations)

Besides the above engineering aspects, we are currently studying processes for automatically extracting knowledge from documents, proactively collaborating with the user on how to use collected information for populating/enriching managed ontologies; we also intend to explore possibilities, requirements and characteristics in realizing a collaborative working environment based on ST, and embrace diverse kind of media sources.

Semantic Turkey site can be reached at: http://semanticturkey.uniroma2.it/

References

1. Adida, B., & Birbeck, M. (2007, October 26). RDFa Primer. Retrieved from World Wide Web Consortium - Web Standards: http://www.w3.org/TR/xhtml-rdfa-primer/
2. Broekstra, J., Kampman, A., & van Harmelen, F. (2002). Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. *The Semantic Web - ISWC 2002: First International Semantic Web Conference* (p. 54-68). Sardinia, Italy: Springer Berlin / Heidelberg
3. Ciravegna, F., Dingli, A., Petrelli, D., & Wilks, Y. (2002). User-system cooperation in document annotation based on information extraction. *13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02.* Springer Verlag
4. DeRose, S., Daniel, R. J., Grosso, P., Maler, E., Marsh, J., & Walsh, N. (2002, August 16). *XML Pointer Language (XPointer).* Retrieved from World Wide Web Consortium - Web Standards: http://www.w3.org/TR/xptr/
5. Dzbor, M., Domingue, J., & Motta, E. (2003). Magpie: Towards a Semantic Web Browser. *2nd International Semantic Web Conference (ISWC03).* Florida, USA.
6. Fallucchi, F., Pazienza, M. T., Scarpato, N., Stellato, A., Fusco, L., & Guidetti, V. (2008). Semantic Bookmarking and Search in the Earth Observation. In I. Lovrek, R. J. Howlett, & L. C. Jain (Ed.), *Knowledge-Based Intelligent Information and Engineering Systems. 12th International Conference, KES 2008, Zagreb, Croatia, September 3-5, 2008, Proceedings, Part III. Lecture notes in Computer Science. 5179/2008*, pp. 260-268. Springer
7. Gennari, J., Musen, M., Fergerson, R., Grosso, W., Crubézy, M., Eriksson, H., et al. (2003). The evolution of Protégé-2000: An environment for knowledge-based systems development,. *International Journal of Human-Computer Studies , 58* (1), 89–123
8. Griesi, D., Pazienza, M. T., & Stellato, A. (2006). Gobbleing over the Web with Semantic Turkey. *Semantic Web Applications and Perspectives, 3rd Italian Semantic Web Workshop (SWAP2006).* Scuola Normale Superiore, Pisa, Italy.
9. Griesi, D., Pazienza, M. T., & Stellato, A. (2007). Semantic Turkey - a Semantic Bookmarking tool (System Description). In E. Franconi, M. Kifer, & W. May (A cura di), *The Semantic Web: Research and Applications, 4th European Semantic Web Conference, ESWC 2007, Innsbruck, Austria, June 3-7, 2007, Proceedings. Lecture Notes in Computer Science. 4519*, p. 779-788. Springer
10. Huynh, D., Mazzocchi, S., & Karger, D. (November, 2005). Piggy Bank: Experience the Semantic Web Inside Your Web Browser. *Fourth International Semantic Web Conference (ISWC05)*, (p. 413-430). Galway, Ireland
11. Kahan, J., & Koivunen, M.-R. (2001). Annotea: an open RDF infrastructure for shared Web annotations. *WWW '01: Proceedings of the 10th international conference on World Wide Web* (pp. 623-632). Hong Kong, Hong Kong: ACM
12. Kiryakov, A., Ognyanov, D., & Manov, D. (2005). OWLIM – a Pragmatic Semantic Repository for OWL. *Int. Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2005), WISE 2005.* New York City, USA
13. *OSGi RFC0112.* (2005). Retrieved from http://www2.osgi.org/Download/File?url=/download/rfc-0112_BundleRepository.pdf
14. Prud'hommeaux, E. :. (2008, January 15). *SPARQL Query Language for RDF.* Retrieved from World Wide Web Consortium - Web Standards: http://www.w3.org/TR/rdf-sparql-query/
15. Raskin, R. (2005). *Semantic web for earth and environmental terminology*. Retrieved from http://sweet.jpl.nasa.gov/index.html