Making the Semantic Web a Reality through Active Semantic Spaces

Yihong Ding[1], Ying Ding[2], David W. Embley[1], Omair Shafiq[2], and Martin Hepp[2]

[1] Department of Computer Science, Brigham Young University, U.S.A.
{ding,embley}@cs.byu.edu

[2] Digital Enterprise Research Institute (DERI), University of Innsbruck, Austria
{ying.ding,omair.shafig,martin.hepp}@deri.org

Abstract. One reason for the slow acceptance of the Semantic Web is the lack of simple, straightforward, and attractive end-user showcases. Interest from end-users is an essential driving force of all web technologies. In this position paper we propose active semantic spaces (ASpaces) as a possible showcase. An ASpace is built on a combination of Semantic Web technologies, service agent technologies, and Web 2.0 technologies. Semantic Web technologies support producing machine-processable content on ASpaces. Service-agent technologies support proactive machine agents on ASpaces that communicate with both their human users and their peers. Web 2.0 technologies support friendly user interaction and the capability of dynamically collecting remote feedback from ASpace agents. By combining these technologies together, users can issue personal requests to ASpace agents, which then look for answers on other ASpaces. ASpace agents automatically blog results back to users as if they had come from remote human users. This showcase is an example of bridging the gap between the Semantic Web and Web 2.0.

1 Introduction

Despite its importance, designing and producing a good Semantic-Web showcase is not easy. To qualify, the showcase must include machine understanding, a nontrivial requirement. To enable machine understanding, we are likely to have to carefully integrate foundation technologies including ontologies, semantic annotation, and semantic search. At the same time, however, we must not make showcases too complex. Satisfying all these requirements is a huge challenge.

In this position paper, we present the idea of Active Semantic Spaces (ASpaces), a potential Semantic-Web showcase, and we explain how, despite the huge challenge, it can be realized. As Figure 1 shows, an ASpace is a combination of three technologies. (1) Semantic Web technologies in an ASpace provide ontology-specified semantics about the ASpace owner and about domains of interest to the owner. (2) Web 2.0 technologies in an ASpace support friendly user interaction and the capability of dynamically collecting remote feedback. (3) Service agent technologies in an ASpace allow machine agents to communicate with each other. Together these technologies support four advances over current web technologies. First, ASpaces provide a new active human communication model between web readers and writers: blog writers can actively find potential readers rather than simply waiting for responses. Second, ASpaces provide query-answering services beyond the capability of current search engines. Third, ASpaces can exist simultaneously with the current web, and the wide adoption of ASpaces may help actualize the dream of the Semantic Web. Fourth, ASpace design is an example of bridging the Semantic Web and Web 2.0.

Our vision of the Semantic Web is close to the vision of Semantic Web 2.0 as discussed by Breslin and Decker [2]. Both of us agree that Web 2.0 is not enough, and we need to add richer semantics into Web 2.0 publications to provide users greater facilities to manipulate web data. The difference is, however, that we emphasize more on the side of enhanced machine communications, while Breslin and Decker focuse more on the side of facilitating human communications. We believe that both sides are crucial to the realization of the next-generation web.

2 ASpaces: A Semantic Web Showcase

A successful showcase must be pragmatic, which means that users can immediately see its value. For example, when a homepage (a traditional web showcase) is created, its developers can display it properly on their own as well as on other computers with internet connections. When a blog (a Web 2.0 showcase) is created, its developer can start to view feedback from readers of the blog immediately. Similarly, when an ASpace is created, its developers must be able to actively find some likely potential readers or desired information.

To make this work and be pragmatic, an ASpace contains ontologies, annotated content, agents, services, and blogging capabilities.

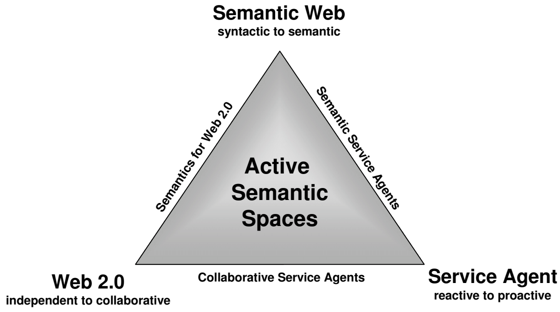

Fig. 1. Origin of Active Semantic Spaces.

2.1 ASpace Ontologies and Annotated Content

The need for ontologies is a major challenge for the Semantic Web. For ASpaces, we believe, as do others [6, 10], in two assumptions: (1) Semantic-Web users need only simple ontologies, and (2) personalization of ontologies simplifies mutual communication between humans and machines.

Based on these beliefs, we have adapted a light-weight ontology representation [4, 5] for ASpace ontologies. ASpace ontologies are equivalent to OWL-DL ontologies in formalism and reasoning power, but they differ from OWL because they include instance recognition semantics.[3] *Instance recognition semantics* are formal specifications that interpret instances of a concept in ordinary text. Syntactically, our declarations of instance recognition semantics allow users to specify recognition phrases, context phrases, and exception phrases in regular expressions for any ontology concept. For example, in a declaration of the concept *Product Price*, we can specify its recognition phrase to be "\d{1,4}(\.\d\d)?," which allows the range of prices to be from 0.00 to 9999.99. Optionally we can add a left-context phrase "\$"[4], context keywords "price" and "product" and an exception context keyword "discount." Therefore, this declaration can correctly recognize the number in "product price: $95.50" to be a legal instance of *Product Price*, and correctly exclude the number in "product discount: $5.00" as a *Product Price*.

Using ASpace ontologies, we can semi-automatically annotate web content. To help annotate web documents, we have developed an automated, ontology-based semantic annotation tool [4],[5] which ultimately is based on the ontology-

[3] We have proposed an extension to OWL that includes instance recognition semantics, and we have implemented conversions to and from OWL and our proprietary ontology language [3].

[4] Right-context phrases are also allowed though our example does not show it.

[5] Online demo is available at http://www.deg.byu.edu/.

based data-extraction technology we have studied and developed for years [5]. The core of this automated semantic annotation tool is the use of instance recognition semantics in ontologies. Our earlier experiments have shown that this semantic annotation tool can automatically annotate data-rich web content with high accuracy [4].

In addition to automated annotation, we also provide a simple manual annotation tool so that users can revise and update automatically created annotations. Users can also use the Web-2.0-style tagging techniques to categorize their documents with respect to selected ASpace ontologies. We store both automatically created and manually created annotated data in RDF files.

2.2 ASpace Blogs and Services

ASpaces are mediators that connect web users to the public web. Each ASpace contains two types of blogs: publication blogs (PuBs) and request blogs (ReBs). Through PuBs, users publish their information to the public. Through ReBs, users issue requests to collect information of interest from the public web. Both PuBs and ReBs are linked to a common local ontology repository, and to the way be annotated. This design is close to the idea of semantic blogs and semantic wikis. Karger and Quan [7], for example, view blogging as a user-friendly way to exchange data and encourage semantic annotation of blogs. Later on, Möller and his colleagues [8] developed a *semiBlog* editor to assist in adding metadata to blog posts. The annotations can be added to individual posts in RDF format through which machines may find connections between different blogs. These approaches, however, have not addressed personalized knowledge specifications and have not enabled users to issue personal requests for active web search.

In ASpaces, PuBs are standard blogs augmented with semantic annotation. Semantic annotations aid remote machine agents by giving them specific information about these PuBs with respect to an ontology. Without annotations, PuBs are assumed to carry only latent information because it is not guaranteed that remote agents can understand web content without annotations. Whether unannotated content can be understood depends on the ability of an agent to automatically annotate the content with respect to a known ontology. PuB owners have complete freedom to decide the percentage and in how much detail they want to annotate their PuBs. Certainly, if the percentage and detail of annotated content is greater, the chance of a PuB may become useful and used by remote agents is greater. This design strategy gives users freedom of choice as well as the motivation to do detailed annotations.

ReBs are private blogs (visible only to the owner) working as communication interfaces between ASpace users and ASpace agents. Users write requests on ReBs and invoke machine agents to understand and execute them. User requests are annotated, and thus aligned with ontologies, by the same automatic and manual annotation module used to annotate PuBs. Once annotated requests become machine-processable, they are converted to executable queries and executed on PuBs at remote ASpaces. After requests are executed, machines can automatically blog the results back on ReBs as if they had come from remote human users.

An ASpace is not only a space that stores user information (PuBs), but also ASpace agents in which machines execute commands and interact with users (ReBs). This is why designing the space as being "active." Since the purpose of Semantic Web is to leverage machine processing, a Semantic Web showcase must contain machine agents. Human readers do not need "machine-processable" content.

We now step through a simple example to show how ASpaces behave and why ASpaces can be a Semantic Web showcase.

Story. Suppose Bob, an ASpace user, likes Nikon Coolpix S5 digital cameras. He wants to become acquainted with people who own or have an interest in this product. He also wants to find coupons for this product for both himself as well as anyone else who might be interested. This story, though simple, has a complicated scenario that cannot be resolved well on the current web. For example, finding people who own or have an interest in Nikon Coolpix S5 is very tedious. Searching for people based on their interests is not well supported on the current web. Searching by product name often results in hundreds of sales, manufacturer, and review pages prior to personal homepages (or blogs) that contain product information. Coupons for a specific product may or may not be easy to find, but even after Bob has found a coupon, it is not easy for Bob to appropriately notify others about this coupon. Since Bob does not really know if others he has found as a result of his search are also interested in these coupons, Bob should avoid broadcasting an email message, which may be received as bothersome spam. ASpaces address all the issues.

Ontologies. This story is about three small domain ontologies: person contact information, digital cameras, and digital coupons. We support initial ASpace creation by providing an array of ontologies including instance recognition semantics for common domains. Ontologies for contact information, digital cameras, and coupons would be among them.

Requests. After Bob has selected ontologies of interest, he can start writing his requests on his ReB. First he writes "Find people who own or have an interest in Nikon Coolpix S5 digital camera." The annotation module in Bob's ASpace annotates this request as follows.

> "Find <person-contact-info:Person>people</...> who own
> or have an interest in <digital-camera:Make>Nikon</...>
> <digital-camera:Model>Coolpix S5</...>
> <digital-camera:Digital Camera>digital camera</...>."

Next, Bob writes "Search for coupons for Nikon Coolpix S5 cameras that remain valid until the end of this month." His annotation module annotates this request as follows.

"Search for <digital-coupon:Coupon>coupons</...> for <digital-coupon:Product Name>Nikon Coolpix S5</...> cameras that remain valid until <digital-coupon:Valid Time>the end of this month</...>."

After Bob has done on requesting, Bob decides that it is the golfing time. So he simply leaves his ASpace agent take care of the rest of the tasks.

Request Execution. Bob's ASpace agent can perform the two requests simultaneously. For the first request, Bob's agent first contacts the ASpace servers to obtain a list of ASpaces that have also downloaded both of the person-contact-information and digital-camera ontologies. Bob's agent enumerates this list and contacts respective remote ASpace agents individually to request checking the annotated content on their PuBs. Once the checking request is granted, it matches the annotated content in the request and in the remote PuB based on common concepts. For example, it checks whether in the remote PuB the annotated *digital-camera:Make* is "Nikon." Once there is a match, it automatically blogs all the annotated record as well as the URL of the respective remote ASpace back to its own ReB. For example, if an agent find a match of "Nikon" and "Coolpix S5" as *Make* and *Model* of digital cameras on Alice's ASpace, it automatically blogs the annotated personal contact information of Alice (*person-contact-info:Person*) to its ReB as well as a link to Alice's ASpace. Execution of the second request is similar.

Wrap Up. Bob has found many new acquaintances who share a common interest and has also some valuable coupons. Bob wants to share coupons with his acquaintances, but he does not want to produce what may perceive to be spam. First Bob clicks on a button to tell his agent to transfer his newly acquired coupon information on the ReB to his PuB. Bob then asks his ASpace agent to send a notice to the agents of these new acquaintances. The notice tells agents at its target that this site knows about coupons for "Nikon Coolpix S5" cameras valid to the end of this month. The notice is stored latently on the remote ASpaces so that their owners would not be annoyed about reading unexpected information. If advocated by their owners, ASpace agents can directly go to Bob's site to get the information rather than first having to contact ASpace servers.

3 Evaluation and Discussion

ASpaces are a potential, attractive showcase for the Semantic Web. ASpaces allow web users publishing everything they can now publish. As an added value, however, ASpaces allow web users to issue many requests that cannot currently be resolved. People have spent too much time digging a little piece of useful information out of tons of uninteresting online publications. By letting ASpace agents execute human requests, people can gain time back and do what they really enjoy.

Our proposed ASpace implementation is pragmatic and flexible. Most of the new core technologies required to build the key components of ASpaces have already been developed: ontologies with instance recognition semantics [5], ontology-based automated semantic annotation [4], automated query generation from annotated user requests [1, 4], and agent communication in the Semantic Web [9]. Although we have decided to use these technologies in our implementation, our ASpace implementation is also open to other technologies. For example, annotation can be presented in either RDF or Microformat and request queries can be formatted with SPARQL or XQuery. As long as they share the same ontologies, different implementations of ASpaces would not prohibit communication among ASpaces.

For end users, setting up ASpaces is the same as setting up blogs, except that now they set up two different types of blogs: PuBs and ReBs. They also choose ontologies from an ASpace server. This is, however, similar to do online bookmarking using del.icio.us, which is now a quite popular and an accepted online human activity. At this point the ASpace is set up, but not as useful as it could be. To make them much more useful, ASpace users can invoke downloaded extraction ontologies to automatically annotate existing blogs and web pages. To make annotated blogs and web pages even more useful, ASpace users can manually correct and add annotations.

What is not straightforward is to construct the ASpace server library of ontologies. Experience shows that it takes a few dozen person hours to construct an ontology for a domain like digital cameras. To aid in this process, we have begun to work on tools to semi-automatically construct ontologies.[6]

4 Concluding Remarks

We emphasize that the major contribution of this ASpace project to the community of the World Wide Web is not its implementation, though this implementation is also a good contribution. It is its design, its philosophy, and its potential impact to the evolution of the web. ASpace is a novel example of bridging the gap among the three parties: from syntactic data display to semantic data description (Semantic Web), from independent web behaviors to collaborative web behaviors (Web 2.0), and from a reactive system to a proactive system (service agent).

5 Acknowledgement

Yihong Ding and David W. Embley are supported in part by NSF grant #0414644. Part of the work presented in this paper is also supported by the European Commission under the projects DIP (FP6-507483), SUPER (FP6-026850), and MUSING (FP6-027097), by the Austrian BMVIT/FFG under the FIT-IT project myOntology (grant no. 812515/9284), and by a Young Researcher's Grant from the University of Innsbruck.

[6] TANGO: Table ANalysis for Generating Ontologies, NSF grant #0414644

References

1. M. Al-Muhammed and D. W. Embley. Ontology-based constraint recognition for free-form service requests. In *Proceedings of 23th IEEE International Conference on Data Engineering (ICDE 2007)*, Istanbul, Turkey, April 2007. (in press).
2. J. Breslin and S. Decker. Semantic Web 2.0: Creating Social Semantic Information Spaces. In *Tutorial in the 15th International World Wide Web Conference (WWW 2006)*, Edinburgh, Scotland, May 2006.
3. Y. Ding, D. Embley, and S. Liddle. Building a bridge between ontologies and automated semantic annotation with OWL-AA. (submitted for review).
4. Y. Ding, D. Embley, and S. Liddle. Automatic creation and simplified querying of semantic web content: An approach based on information-extraction ontologies. In *Proceedings of the First Asian Semantic Web Conference (ASWC 2006) LNCS 4185*, pages 400–414, Beijing, China, September 2006.
5. D. Embley, D. Campbell, Y. Jiang, S. Liddle, D. Lonsdale, Y.-K. Ng, and R. Smith. Conceptual-model-based data extraction from multiple-record web pages. *Data & Knowledge Engineering*, 31(3):227–251, November 1999.
6. A. Iskold. The road to the Semantic Web. http://www.readwriteweb.com-/archives/the_road_to_the.php.
7. D. Karger and D. Quan. What would it mean to blog on the Semantic Web? In *Proceedings of the Third International Semantic Web Conference (ISWC 2004)*, pages 214–228, Hiroshima, Japan, November 2004.
8. K. Möller, J. Breslin, and S. Decker. semiblog—semantic publishing of desktop data. In *Proceedings of the 14th Conference on Information Systems Development (ISD 2005)*, Karlstad, Sweden, August 2005.
9. O. Shafiq, Y. Ding, and D. Fensel. Enabling interoperability between information systems by bridging multi agent systems and web services. In *Proceedings of 10th IEEE International Enterprise Computing Conference (EDOC 2006)*, Hong Kong, China, October 2006.
10. N. Spivack. What is the Semantic Web, actually? http://novaspivack.typepad.-com/nova_spivack_weblog/2006/11/what_is_the_sem.html.