

Demo Abstract: Bringing the Semantic Web to the Office Desktop

Timothy Miles-Board
tmb@ecs.soton.ac.uk

Arouna Woukeu
aw1@ecs.soton.ac.uk

Leslie Carr
lac@ecs.soton.ac.uk

Gary Wills
gbw@ecs.soton.ac.uk

Wendy Hall
wh@ecs.soton.ac.uk

Intelligence, Agents, Multimedia Group, University of Southampton, UK

ABSTRACT

Many Semantic Web applications address the needs of human readers of the Web (e.g. searching, annotating), but these technologies can also address the needs of human writers of the Web. The WiCK project has explored the application of knowledge bases and services to the Office desktop, in order to assist document production, culminating in the WiCKOffice environment. This aim of this demonstration is to showcase the most recent offshoot of the WiCKOffice development, WiCKLite: a lightweight component for connecting knowledge services to document templates in order to deliver targeted assistance to end users.

Categories and Subject Descriptors: I.7.2 [Document and Text Processing] : Document Preparation

General Terms: Human Factors

Keywords: Semantic Web, Knowledge Writing, Smart Tags

1. INTRODUCTION

The aim of the Semantic Web is to make the present web more machine-processable, in order to allow intelligent agents to better retrieve and manipulate information. The end client is usually a human, rather than machine, intelligent agent. Many Semantic Web applications aimed at humans have been concerned with searching [4] and annotating [1, 3]. Both these types of application address readers of the Web, but Semantic Web technologies can equally well address the needs of writers, particularly those who need to synthesise information from various sources in order to create new documents.

At the 2004 Document Engineering conference we presented a case for embedding explicit knowledge in documents [2], which supported the notion that semantic annotation of new documents should be an essential and active component of the writing process itself, rather than left to fallible, post-hoc extraction processes. We introduced the Writing in the Context of Knowledge (WiCK) project¹, which investigated the application of Semantic Web technologies in an office environment (WiCKOffice). So as to make such an investigation practical, the work focused on

¹<http://wick.ecs.soton.ac.uk/>, EPSRC GR/R91021/01

a specific kind of writing task: producing a funding proposal. By orchestrating a number of services operating over an existing knowledge store² via extensions to the Microsoft Office suite, WiCKOffice was able to provide assistance in several facets of this task.

Of these services, the *Knowledge Recall* service forms the basis of the work demonstrated here. Using a smart tag interface, this service delivered targeted knowledge into the document in the *right place* (i.e. appropriate to the section of the document currently being worked on) at the *right time* (when the name of a recognised person, institution, or project was typed by the author). This approach has been extracted and generalised as *WiCKLite* — a lightweight smart tag component which can be used to wire up existing (knowledge) services to writing tasks using a straightforward XML syntax.

2. WICKLITE DEMONSTRATION

Developers have been able to easily create, deploy and manage simple smart tags — using the Microsoft Office Smart Tag List (MOSTL) XML syntax to specify the terms to recognise and associated actions to present to the user³ — since Office XP (2001). However, the MOSTL syntax lacks some key features which make it unsuitable for deployment in knowledge-based environments: association of URIs (i.e. a reference to a concept in an ontology) with text terms, delivery of knowledge directly into the document, and the display of different actions for a single term, depending on document context. WiCKLite leverages the simplicity of the MOSTL syntax, but extends it to address these shortcomings, allowing developers to specify a URI for each term the smart tag will recognise, whether the result of an action is to be displayed in the browser or inserted directly into the document, and the document contexts to which an action should apply (using regular expressions). WiCKLite therefore offers a highly configurable approach for individuals, communities, and organisations to increase their document productivity without sacrificing existing Microsoft Office applications and skill-sets.

We envision a typical WiCKLite deployment scenario within an organisation as being a lightweight “middleware” component connecting Web services (operating over a corpo-

²<http://triplestore.aktors.org>

³http://msdn.microsoft.com/library/en-us/dnsmarttag/html/odc_stxml.asp

rate database or knowledge-base) with in-house templates for producing documentation, delivering timely information and/or knowledge directly to workers as they produce and/or consume documents. To briefly outline how WiCKLite can be deployed in such a scenario, this demonstration shows how an Academic CV template marked up with an XML Schema⁴, together with a WiCKLite configuration describing Semantic Web services operating over the a knowledge-base (Listing 1) can be combined to deliver targeted assistance to end users as they create their own Academic CVs.

In Listing 1, a smart tag type `Person` is declared. The `termfile` element specifies that the list of terms to recognise (and their associated URIs, since the `instances wick` is set) should be downloaded from the specified URL. Two actions are then specified for these terms, which are made available to the user in different document contexts.

Listing 1: Extract of WiCKLite configuration for writing an Academic CV (WiCK extensions to the MOSTL syntax highlighted).

```
<smarhtag type="urn:wick.ecs.soton.ac.uk:tags#Person">
  <caption>Soton AKT People</caption>
  <terms wick:instances="true">
    <termfile>
      <filename>http://wick.ecs.soton.ac.uk/wicklite/ecs-
        people
      </filename>
    </termfile>
  </terms>
  <actions>
    <action id="PersonHomepage" wick:behaviour="
      ShowInBrowser">
      <caption>Visit Person Homepage(s)</caption>
      <url>http://wick.ecs.soton.ac.uk/wicklite/
        haswebaddress?resource=%URI&label=%TERM
      </url>
      <wick:contexts>
        <wick:context>academic-cv:Research</wick:context>
        <wick:context>academic-cv:* Activities
      </wick:contexts>
    </action>
    <action id="PersonPubs" wick:behaviour="
      InsertIntoDocument">
      <caption>Insert Most Recent Publications
      </caption>
      <url>http://wick.ecs.soton.ac.uk/wicklite/
        hasmostrecentpubs?resource=%URI&label=%TERM
      </url>
      <wick:contexts>
        <wick:context>academic-cv:Publications
      </wick:contexts>
    </action>
  </actions>
</smarhtag>
```

If a term is recognised within those areas of the document marked up with the `Research` (including its subelements `Current`, `Previous` and `Achievements`), `AcademicActivities` or `CommunityActivities` elements of the Academic CV schema, the user will be able to visit the homepage of the recognised person (Figure 1). The `haswebaddress` service consults the knowledge-base and determines the homepages associated with the given URI, returning a list that can be displayed in a browser.

However, if the term is recognised within that part of the document which has been marked up with the `Publications` element of the Academic CV schema, the option to insert that person's most recent publications will be made available (Figure 2). The `hasmostrecentpubs` service queries the knowledge-base for the most recent publications associated with the given URI, and renders the list as a WordML⁵ fragment which can be inserted directly into the document.

⁴http://msdn.microsoft.com/library/en-us/odc_wd2003_ta/html/odc_wdxmlresumetemp.asp

⁵<http://msdn.microsoft.com/library/en-us/dnofftalk/html/office01022003.asp>

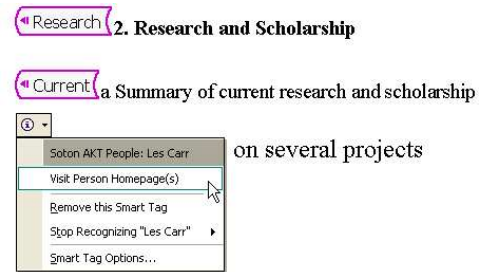


Figure 1: Actions available on the recognised term "Les Carr" in the region marked up with the `Current` element.

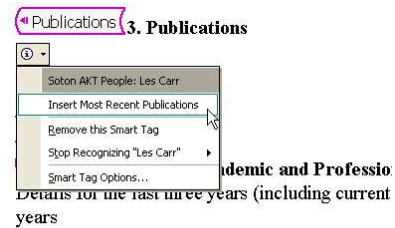


Figure 2: Actions available on the recognised term "Les Carr" in the region marked up with the `Publications` element.

Services can take advantage of the WordML syntax and Semantic Web languages such as RDF to produce formatted output with explicit embedded semantics (which can later be harvested).

3. CONCLUSION

Properly harnessed, the Semantic Web has the potential not only to streamline the existing writing process, but also to help authors improve the coherence and consistency of their written output: work in this area therefore potentially impacts every sector of industry and commerce concerned with document and information management. We hope that our lightweight WiCK component will encourage experimentation with combining knowledge-aware services and document templates in order to deliver targeted assistance (and embedded knowledge) to writers of the Semantic Web, and help realise this potential.

4. REFERENCES

- [1] Leslie Carr, Sean Bechhofer, Carole Goble, and Wendy Hall. Conceptual linking: Ontology-based open hypermedia. In *Proceedings of the Tenth International World Wide Web Conference, Hong Kong, 2001*.
- [2] Leslie Carr, Timothy Miles-Board, Arouna Woukeu, Gary Wills, and Wendy Hall. The case for explicit knowledge in documents. In *Proceedings of the ACM Symposium on Document Engineering (DocEng2004), Milwaukee, WI, USA, pages 90-98, 2004*.
- [3] Martin Dzbor, John Domingue, and Enrico Motta. Magpie: towards a semantic web browser. In *Proceedings of the 2nd International Semantic Web Conference, Sanibel Island, FL, USA, pages 690-705, 2003*.
- [4] Jeff Heflin, James Hendler, and Sean Luke. Reading between the lines: Using SHOE to discover implicit knowledge from the web. In *Proceedings of the AAAI-98 Workshop on AI and Information Integration, Madison, WI, USA, 1998*.