3.B. ANÁLISIS Y RECUPERACIÓN DE INFORMACIÓN

Esta sección incluye los siguientes temas:
Metadatos, vocabularios, thesaurus, ontologías, web semántica, procesamiento del lenguaje natural, buscadores, catalogación.

B.1. Navegadores semánticos o semantizar el navegador

Por **Jose A. Senso**

Senso, Jose A. "Navegadores semánticos o semantizar el navegador".
En: *Anuario ThinkEPI*, 2008, pp. 30-33.

Resumen: *Un navegador semántico es cualquier herramienta que permita visualizar contenido semántico, mientras que semantizar un navegador consiste en añadirle capacidades extra para que pueda mostrar esa información. La gran duda del mercado en la actualidad reside en determinar qué camino escoger para desarrollar los navegadores de la Web semántica*.

Palabras clave: *Web semántica*, *navegador semántico*, *contenido semántico*, *BigBlogZoo*, *Tabulator*, *proyecto Haystack*, *Piggy Bank*, *mSpace*.

Title: Semantic browsers or making browsers semantic

Abstract: *A semantic browser is any tool for browsing semantic content, or for browsing content based on semantic data. Other systems exist that let users make use of Semantic Web content within Web content as they browse the Web. The great unknown at present is how the market will decide which path to choose in developing browsers for the Semantic Web*.

Keywords: *Semantic Web*, *Semantic browser*, *Semantic content*, *BigBlogZoo*, *Tabulator*, *Haystack project*, *Piggy Bank*, *mSpace*.

TRANSCURRIDOS VARIOS AÑOS desde la aparición del concepto de Web semántica parece que, en determinados aspectos, el "invento" está evolucionando de forma más que favorable. Existe gran cantidad de proyectos que emplean alguna (o casi todas) las capas del famoso gráfico explicativo[1] de Berners-Lee. En la actualidad se pueden encontrar muchos programas que, con mayor o menor éxito, han logrado plasmar muchas de las ideas introducidas en esta filosofía de gestionar los datos.

Pero, donde todavía se han dado pocos pasos –o al menos no son muy firmes– es en la práctica más visible para los usuarios: los navegadores. Se supone que toda esa gran cantidad de información que estará estructurada en xml, descrita con metadatos, organizada gracias a las ontologías y recuperada por medio de los agentes inteligentes, debería ser visible por algún método. żY qué mecanismo es con el que el usuario medio está más familiarizado e integrado dentro de la actual internet?

Efectivamente, los navegadores sirven de unión entre el internauta y la información, obviando y haciendo transparente todo ese conglomerado de siglas, protocolos y normas. Si esto sucede en la Red con la que se trabaja hoy en día, independientemente de si tiene el apellido 2.0, dinámica, blogosfera

En lo más visible para los usuarios, los navegadores, la Web semántica ha dado pocos pasos –o al menos no son muy firmes–

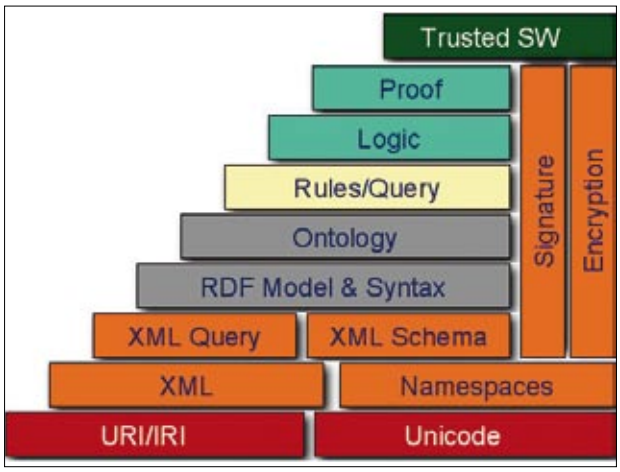

Gráfico de Tim Berners-Lee sobre la web semántica http://en.wikipedia.org/wiki/Image:W3c_semantic_web_stack.jpg

No se puede decir que los proyectos que opten por crear navegadores semánticos específicos tengan una base más sólida que los que se decidan por extender las posibilidades semánticas de los navegadores actuales

o el nuevo que se quieran inventar, lo lógico es que algo sucedido pase con la evolución natural: la Web semántica.

Un rápido recorrido por los proyectos y, especialmente, por el software semántico lleva a distinguir entre dos métodos diferentes para realizar navegadores que permitan visualizar ese tipo de información. La primera de ellas, "navegadores semánticos", es la que pretende hacer *browser* pensados especialmente para este caso. La segunda, "semantizar el navegador", añade determinados elementos a los actuales programas para extender sus posibilidades de navegabilidad aprovechando determinadas características semánticas añadidas a las páginas web.

En el grupo de los "navegadores semánticos" destaca, cómo no, una invención de **Tim Berners-Lee,** que no está teniendo tanto éxito como debiera. *Tabulator*[2] es su nombre y, aunque en estas primeras versiones trabaja como un navegador dentro de otro, lo lógico es que con el paso del tiempo la evolución lo convierta en un software independiente. Este programa en código abierto, basado en *Ajax*, funciona dentro de *Firefox* (es necesario solventar un pequeño problema de seguridad tal y como se explica en su ayuda)[3] o como *widget* de *Opera*[4].

Este navegador se basa en un protocolo que sus creadores han llamado "migas de pan" (*breadcrumbs*). La idea es navegar por recursos de forma escalable, lo que supone que no necesita cargar en memoria toda la información contenida dentro del fichero que está visualizando (generalmente en *RDF*, pero no de forma excluyente). La información se va mostrando conforme el usuario la va necesitando, al igual que una persona va recogiendo poco a poco migas de pan del suelo para llegar al destino deseado.

Junto a este nuevo sistema, el navegador permite identificar en un mapa la localización geográfica expresada en un fichero (por ejemplo, quien tenga identificado en un fichero *Foaf*[5] las coordenadas de su lugar de trabajo, puede mostrar su ubicación exacta) por medio de *Google Mashup*[6] o realizar consultas empleando el lenguaje de búsqueda sobre *RDF Sparql*[7]. Aunque todavía le queda mucho camino que recorrer, la plataforma propuesta es bastante prometedora. Por supuesto, existen otros navegadores dentro de esta categoría que son capaces de mostrar información semántica de manera similar. Entre ellos, destacan *BigBlogZoo*[8], el cliente *Haystack*[9] que funciona sobre *Eclipse*[10] o *Aktive Space*[11].

Aunque existen muchos programas que permiten "semantizar el navegador", el que más se está extendiendo y más posibilidades de futuro presenta es *Piggy Bank*[12]. Se trata de una extensión escrita en *Java* para *Firefox* que permite extraer determinados elementos clave de una página web y almacenarlos en *RDF*.

Dependiendo de la información que encontremos, *Piggy* actuará de dos maneras diferentes. Así, si el sitio tiene un fichero *RDF* o cualquier aplicación de éste, como *Foaf*, o metainformación independientemente de si es *Dublin Core* o metaetiquetas de html, el programa capturará esa información y la integrará en un repositorio, a modo de base de datos local, organizada en función de la estructura descrita. Si, por el contrario, el sitio no dispone de información de este tipo,

el software invocará a un *scraper* para que extraiga esta información y la estructure.

El *screen scraping* es una técnica que se emplea para la extracción automática de texto, obviando la información binaria (imágenes, multimedia, etc.). Los *scrapers* son programas capaces de trabajar con cualquier texto para procesarlo y estructurarlo. De hecho, son muy empleados por los buscadores de internet como anexo al trabajo realizado por sus arañas. *Scroogle*[13], por ejemplo, utiliza esta técnica para hacer búsquedas en *Google* sin que salgan los molestos anuncios alrededor de los resultados.

Piggy incluye tres *scrapers* diferentes escritos en *JavaScript* que son totalmente configurables –sólo hay que tener unos conocimientos mínimos en este lenguaje de programación– pero, además, se pueden emplear nuevos pensando en recuperar imágenes en *Flickr*[14] (*FlickrPhotoScraper*)[15] o búsqueda de amistades para activar redes sociales o *Orkut Friends Scraper*[16] o *LinkedIn*[17]). Incluso explica cómo hacer uno para realizar búsquedas de apartamentos[18].

A la información recogida se le pueden añadir etiquetas para describirla. La técnica de contribuir cada uno poniendo palabras clave se ha hecho muy popular gracias a sitios como *del.icio.us*[19] o *CiteULike*[20], ya que permite que una comunidad construya una taxonomía y publicarla en un banco semántico global, y ésta es otra de las opciones interesantes que observamos en *Piggy Bank*. El banco semántico[21] es un repositorio comunitario de descripciones realizadas en *RDF* que permite a sus usuarios compartir la información que han recogido. Es un mecanismo muy sencillo de publicar y compartir información estructurada. Aunque en la actualidad sólo hay dos: uno genérico[22], que es un caos; y otro específico creado para el congreso *Iswc 2005*[23], la idea de poder crear bancos semánticos para grupos profesionales, por áreas temáticas, etc. es más que interesante. No deja de serlo menos estudiar un mecanismo que permita aglutinar todas las etiquetas creadas individualmente por los usuarios del sistema. Si una ontología fuese capaz de recoger los nombres aportados en una folksonomía y permitiese que, además, la gente pudiera definir las relaciones entre ellos, se facilitaría la creación de *folksolo-*

Figura 1. Éste es el aspecto que tiene un recurso una vez descrito por Piggy Bank. *La mayoría de los elementos para hacer la descripción se extraen automáticamente de los metadatos de las páginas o son creados por los* screenscrapers. *Además, es posible añadirle etiquetas propias para conseguir una organización de los recursos más personalizada.*

gies (*folk ontologies*). Pero esto ya es tema para otro texto.

Además de *Piggy* existen otros programas, que se presentan como extensiones de *Firefox* y que permiten ampliar las posibilidades del navegador. De todos ellos destacan especialmente *Greasemonkey*[24] y *Chickenfoot*[25], ya que facilitan la inclusión de *scripts* para manipular elementos de las páginas web de forma automatizada.

En realidad no se puede cerrar este texto con una conclusión. No se puede decir que los proyectos que opten por crear navegadores semánticos específicos tengan una base más sólida que los que se decidan por extender las posibilidades semánticas de los navegadores actuales. A lo mejor un sistema híbrido, que contemplase el protocolo de migas de pan de *Tabulator*, con la generación automática de descripciones *RDF* y el banco semántico de *Pi-*

ggy Bank, junto con las capacidades de navegabilidad y búsqueda de *mSpace*[26] (por cierto, que tras ver la demo[27] a cualquiera se le ocurren multitud de posibilidades de aplicar ese programa a una biblioteca) conformaría el navegador ideal.

Notas

1. *http://en.wikipedia.org/wiki/Image:W3c_semantic_web_stack.jpg*

2. *http://www.w3.org/2005/ajar/tab*

3. *http://dig.csail.mit.edu/2005/ajar/ajaw/Help.html#_Security*

4. *http://widgets.opera.com/widget/5053*

5. *http://www.foaf-project.org/*

6. *http://www.thinkepi.net/repositorio/nuevas-formas-de-vida-en-la-web-mashups-bibliotecarios/*

7. *http://www.w3.org/TR/rdf-sparql-query/*

8. *http://www.bigblogzoo.com/*

9. *http://haystack.csail.mit.edu/home.html*

10. *http://www.eclipse.org/*

11. *http://triplestore.aktors.org/SemanticWebChallenge/*

12. *http://simile.mit.edu/piggy-bank/*

13. *http://scroogle.org/*

14. *http://www.flickr.com/*

15. *http://simile.mit.edu/wiki/FlickrPhotoScraper*

16. *http://simile.mit.edu/wiki/Orkut_Friends_Scraper*

17. *http://simile.mit.edu/wiki/LinkedIn_Scraper*

18. *http://simile.mit.edu/piggy-bank/screencasts/apartments.swf*

19. *http://del.icio.us/*

20. *http://www.citeulike.org/*

21. *http://simile.mit.edu/wiki/Semantic_Bank*

22. *http://simile.mit.edu/bank/*

23. *http://simile.mit.edu/conference/iswc2005/*

24. *http://www.greasespot.net/*

25. *http://groups.csail.mit.edu/uid/chickenfoot/*

26. *http://www.mspace.fm/*

27. *http://beta.mspace.fm/*