

Herramientas para trabajar con *rdf*

Por Jose A. Senso

RESULTA EVIDENTE que las estructuras de metadatos están adquiriendo una posición preponderante en lo que se refiere a la descripción de recursos electrónicos entendidos como objetos. Cada vez son más numerosos los proyectos, sitios web o sistemas de consulta que se valen de ellos para lograr mejores prestaciones a la hora de la representación, localización y recuperación de recursos electrónicos.

Al contrario de lo que sucede con formatos de metadatos más complejos y menos flexibles como

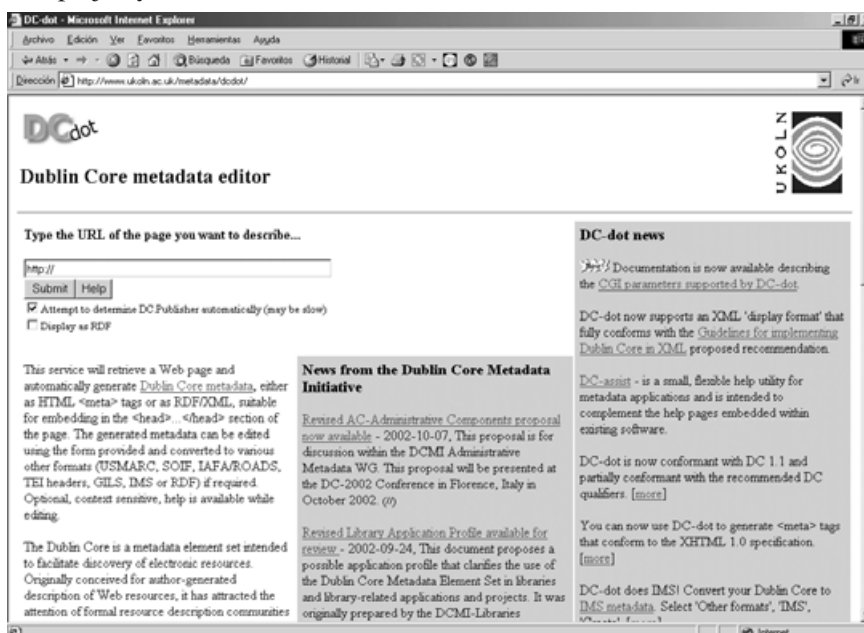
TEI (*text encoding initiative*), el sistema más extendido en la actualidad —*Dublin core metadata initiative* (*Dcmi* en adelante)— pone más énfasis en facilitar al máximo el acceso al recurso y menos en proporcionar una descripción exhaustiva del mismo. Esto resulta vital, ya que ha habido un fallo tradicional en los catálogos bibliotecarios en los que, por el contrario, se hace más hincapié en la descripción que en dotar a los registros de más y mejores elementos para la recuperación.

Junto a *Dcmi* destaca sobre manera *rdf* (*resource description framework*), un sistema de metadatos que sirve tanto para la descripción por sí mismo de recursos electrónicos, como para envolver otros sistemas de metadatos con el fin de lograr un marco genérico de trabajo, facilitando de esta forma la interconexión entre diferentes métodos de descripción. En este trabajo analizaremos, de manera breve, algunas características de este sistema así como las principales herramientas que podemos encontrar en la Red para gestionar ficheros *rdf*.



Resource description framework

Rdf fue creado en agosto de 1997 bajo los auspicios del *World Wide Web Consortium* (en adelante *W3C*) con el fin de tener un formato que permitiera alcanzar la compatibilidad entre los diversos sistemas de metadatos, suministrando para ello una arquitectura genérica de metainformación. Por ese motivo se decidió utilizar el lenguaje xml como sistema de comunicación.



El primer borrador público data del 2 de octubre de 1997 (en agosto fue cuando se reunió por primera vez el grupo de trabajo que se encargaría del desarrollo del formato) y, tras diferentes esbozos, correcciones y propuestas, el 17 de febrero de 1999 aparece la última versión publicada como recomendación W3C.

En palabras de **Hjelm (Hjelm, Johan, 2001)**, *rdf* es un formato que tiene como origen dos ramas recientes de la Documentación. Por un lado se encuentran los metadatos —al ser éste un sistema que, además de servir como modelo de metadatos, es capaz de interconectar sistemas entre sí— y, por otro, de la representación del conocimiento —encarnada ahora en el novedoso concepto de *semantic web*—.

La capacidad que posee *rdf* para procesar metadatos facilita la interoperabilidad entre diversas aplicaciones, proporcionando un mecanismo perfecto para el intercambio de información a través del web. Tal y como se afirma en la recomendación W3C, *rdf* tiene distintas áreas de aplicación; como la recuperación de recursos (proporcionando mejores prestaciones a los motores de búsqueda), la catalogación en bibliotecas digitales (especificando también las relaciones de contenido disponibles en un sitio web determinado), los agentes

University of Essex
Information Systems and Software Techniques

[RDF Survey] [Staff] [Education] [Research and Development] [Interesting Resources]

XWMF - An eXtensible Web Modeling Framework

By means of the XWMF (eXtensible Web Modeling Framework) project we want to evaluate the suitability of RDF as a basis for modeling web based information systems. XWMF is based on the Resource Description Framework (RDF) [RDF Survey]. Both frameworks, RDF and XWMF use metadata to describe properties of web resources. Our proposed framework applies the same (meta-) data model to specify the structure and the content of a web site and to make statements about the elements of a web site. A formal model defining the various properties can lead to a consistent structure and easier maintenance. As RDF is the basis, a XWMF model results in machine-understandable and inter-operable content- and meta-data of a web site. The data model is capable to define the structure of information from small units such as elements or single web pages (for authoring-in-the-small) to large structures such as web sites (for authoring-in-the-large). The formal model of XWMF can lead to the development of better software tools supporting web site development and maintenance. Outcomes of the XWMF project we want to present are the XWMF Tools (a collection of XOTe and Prolog programs for processing RDF data) and the GramToR (a graphical RDF editor). The XWMF Tools (Tcl) are available as open source. See below for detailed copyright information. The programs are developed and tested with various distributions (Red Hat and SuSe) of Linux. Further we developed the XWMF Tools (Prolog) to experiment with A Logical Interpretation of RDF (in first order logic, FOL) and with the XRDF approach. XRDF provides a very simple XML-Syntax for the description of a nested triple model. The relation of this approach to RDF is discussed and it is argued, that most of the perceived deficiencies of RDF are non-issues in the context of XRDF.

- [Further Readings](#)
- [XWMF Tools \(Prolog\)](#)
- [XWMF Tools \(Tcl\)](#)

inteligentes (facilitando el intercambio de conocimiento), en sistemas de gestión de propiedad intelectual (expresando políticas de privacidad de un determinado objeto)... (**Brickley, Dan y Guha, R. V., 2000**).

Tras un análisis detallado de la recomendación, así como de la documentación generada por autores afines al W3C, podemos concluir que los siguientes elementos pueden formar parte de una posible definición de *rdf*:

—Sistema que permite la interoperabilidad entre aplicaciones mediante el intercambio de información legible por ordenador a través del web (**Brickley, Dan y Guha, R. V., 2000**).

—Mecanismo que facilita la automatización de procesos susceptibles de ser realizados con recursos web (**Lassila, Ora, 1997**).

—Infraestructura que permite la codificación, intercambio y reutilización de metadatos estructurados (**Miller, Eric, 1998**). Es capaz, además, de fusionar diferentes sistemas de metadatos utilizados para la descripción de recursos web (**Iannella, Renato, 1998**).

—Es una forma de expresar relaciones entre objetos (**Hjelm, Johan, 2001**).

El objetivo fundamental de *rdf* se centra en establecer un mecanismo que permita describir recursos —entendidos éstos como objetos— que tengan como principios la multiplataforma (es decir, independencia de software y/o sistema operativo) y la interoperatividad de metadatos (que posibilite fusionar diferentes descripciones de recursos realizadas con distintos conjuntos de metadatos). El mecanismo utilizado para ello debe ser neutral con respecto al área de aplicación y, al mismo tiempo, lo suficientemente flexible como para describir cualquier tipo de información.

En cierto modo parece lógico que *rdf* naciera en el seno del W3C, ya que esta institución siempre mostró una relación especial con todo lo que tiene que ver con metadatos. De hecho, podemos afirmar que el grupo de trabajo que desarrolló *rdf* se nutrió de iniciativas de otros colectivos:

—Para empezar, hay que reseñar las actividades desarrolladas por un destacado número de empresas (*Blue Angel, Verity, IBM, Oracle*) y el impulso dado por los creadores de los principales navegadores: *Netscape* (*Mozilla* es capaz de leer *rdf* y de gestionar bookmarks basándose en él) y, en menor medida, *Microsoft Amaya*, el navegador web del *W3C*, permite añadir descripciones personalizadas a páginas web y a imágenes.

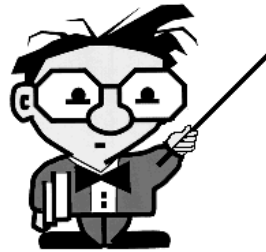
En 1995 apareció *Pics* (*Platform for internet content selection*), una infraestructura que fue originalmente diseñada para ayudar a padres y docentes a controlar el acceso de niños a internet mediante la asociación de etiquetas html, pero que tiene, además, otros campos de actuación, entre los que destaca la protección de los derechos de autor y de propiedad intelectual.

¿Por qué rdf?

Rdf ofrece la solución técnica necesaria para realizar una descripción homogénea y estricta de los recursos sin necesidad de limitar otras opciones, como las de localización y recuperación.

Al mismo tiempo, la posibilidad de utilizar *rdf* junto a cualquier lenguaje de marcas derivado del *sgml* le aporta la característica de multiplataforma, lo que lo convier-

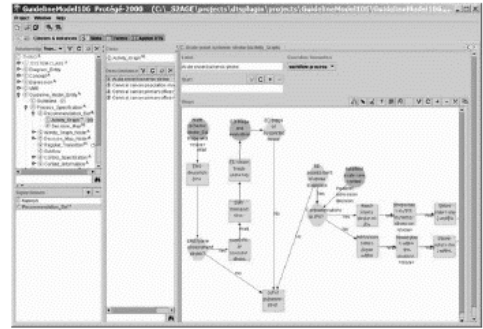
Welcome to the Protégé Project



Protégé-2000 is an ontology editor and a knowledge-base editor.

Protégé-2000 is also an open-source, Java tool which provides an extensible architecture for the creation of customized knowledge-based tools.

Release 1.7 April 10, 2002
Beta 1.8 December 16, 2002



Protégé Community Statistics	
Registered Users	6117
users list members	3556
discussion list members	1194
discussion list messages	3226
Plug-ins	35

Updated January 02, 2003

te en la herramienta ideal para crear un entorno de información integrada en el que el catálogo proporcione acceso tanto a los documentos tradicionales como a la información electrónica.

El uso de conjuntos de metadatos que faciliten la interoperatividad entre diversas bases de datos (como puede ser el uso de *rdf* y *Dc-mi*), la utilización de lenguajes de etiquetado más manejables que el *sgml* y menos simples que el *html* (*xml*), la aplicación de protocolos pensados para la recuperación de información (*Z39.50*), el empleo de técnicas de recuperación de información para generar servicios determinados (*DSI*, por ejemplo)

así como el desarrollo del web, hacen vislumbrar un futuro halagüeño a este sistema de metadatos que tiene como principales características el ser:

- neutral (al no estar ligado explícitamente a ningún otro sistema de metadatos),
- expresivo (las etiquetas son muy intuitivas y se adivina fácilmente su posible contenido),
- familiar (su base *sgml* lo convierte en asequible para las personas relacionadas con *html*), y
- simple de procesar (al ser texto *ascii*).

Por último no se puede obviar que detrás de *rdf* se encuentra una institución como el *Consortio W3*, lo que le confiere cierto rango de estándar de facto.

Por todos estos motivos ha aparecido en los últimos años gran cantidad de aplicaciones que facilitan el trabajo con este sistema de metadatos. A continuación mencionaremos los más interesantes y utilizados.

1.- Apis y bibliotecas

La posibilidad de contar con herramientas que sean capaces de procesar y gestionar ficheros *rdf* es

Metadata Editor

Reggie - The Metadata Editor

The aim of the Reggie Metadata Editor is to enable the easy creation of various forms of metadata with the one flexible program. As it stands, the Reggie applet can create metadata using the HTML 3.2 standard, the HTML 4.0 standard, the RDF (Resource Description Framework) format and the RDF Abbreviated format.

The Reggie Metadata Editor uses a schema file to read in the details of all the elements in a set, their characteristics and descriptions. To create metadata based on a different element set or different language, one has simply to create a new schema file.

The version of Java Security used by Web browsers does not allow copy and paste between applet windows and other windows at the current time. The next version is expected to allow this. Reggie provides two work-arounds to this limitation. If the metadata is small in size, it can be exported to a web page. Otherwise, Reggie can email its results to the user if they provide their email address. WARNING: Some firewalls will block access for email from Java applets. If this is the case, consult your site administrators to allow metadata@net through your firewall.

NOTE: This applet requires use of the latest browser from either Netscape or Microsoft. It loads and runs faster using the Netscape browser.

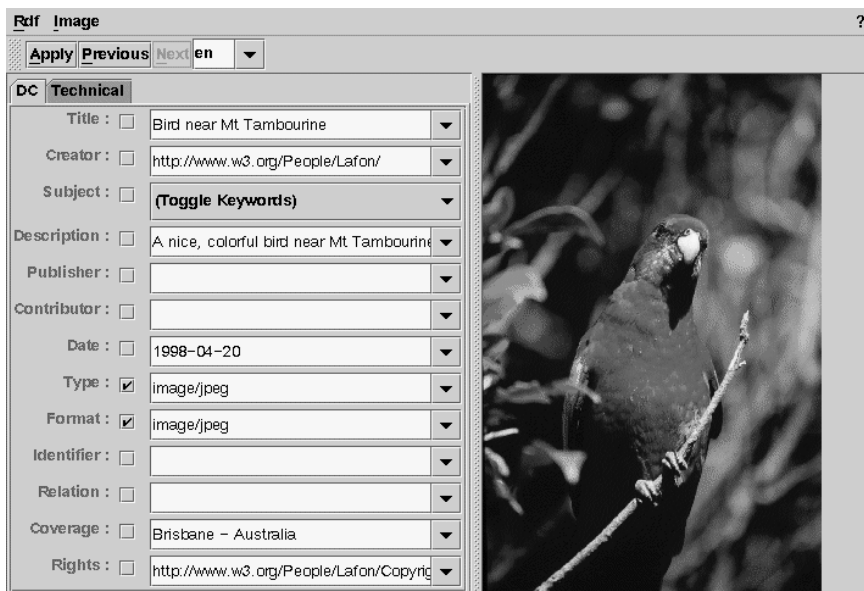
Reggie provides for the Metadata Element Sets and languages listed below. You can use your own schema by entering the URL of the schema directory in the space provided. For information on creating and using your own schema file with Reggie, please refer to [Creating your own schema files](#).

If the resource to be described has a web page, enter the URL. Reggie will extract the META tags from the page and attempt to add them to the most appropriate fields for the chosen schema. For example, if you choose the Dublin Core schema and the page contains a GILS Title META tag, this metadata will be added to the Dublin Core Title field.

Please choose a schema and language:

Schema: Language:

Or, enter the URL for another schema:



muy importante, ya sea desde el punto de vista de la creación o de la consulta.

En la mayoría de los casos, la realización de una *api* (*application programming interface*) que se encargue de estas tareas es la solución más socorrida. Por lo general, la *api* es un módulo que se realiza de forma rápida (suelen usar lenguajes script) y que es fácilmente integrable dentro de cualquier programa ya terminado.

El principal problema radica en que no existe una *api* estándar para *rdf*. Sí podemos contar con numerosas *apis* para xml, pero teniendo en cuenta que *rdf* es sólo un subconjunto creado de xml, es posible que estas *apis* sólo permitan transportar y codificar ficheros.

Actualmente se pueden encontrar algunas *apis* más o menos flexibles realizadas en *java* o *perl*, pero ninguna de ellas sirve como modelo para la creación de una *api* estándar. Entre ellas destacan las siguientes:

—*Cara*: escrita en *perl*, es producto del trabajo de investigación del grupo *Carmen* (*Content analysis retrieval and metadata: effective networking*). Permite almacenar gráficos *rdf* de forma eficiente, pero en su última versión no se han

desarrollado aún las posibilidades de búsqueda.

<http://cara.sourceforge.net/>

—*DataCraft*: es una aplicación pensada fundamentalmente para el comercio electrónico. En esencia lo que hace es visualizar el contenido de bases de datos con registros *rdf* o xml para realizar, posteriormente, consultas sobre ellos utilizando *DB2* —el sistema de bases de datos relacional de *IBM*—, o *Microsoft Access*.

<http://www.alphaworks.ibm.com/tech/DataCraft>

—*Jena*: se trata de una *api* experimental realizada por *Hewlett Packard* en *java* para manipular modelos *rdf*. Hay que destacar que permite trabajar con contenedores y que, además, incluye un parser sintáctico. Por último, es capaz de procesar información *RSS* (*rdf site summary*), lo que la convierte en un módulo ideal para cualquier proceso automatizado de creación de contenidos destinados a ser utilizados en canales de información. <http://www.hpl.hp.com/semweb/download.htm>

—*Rdf crawler*: es una *api* realizada en *java* y pensada para trabajar de forma independiente o conjuntamente con otro programa. Este módulo funciona especialmente en *Windows*, por lo que hay

que tener instalado *JRE* (*java runtime environment*) o *JDK* (*java developer kit*) para ejecutarlo. Incluye el parser *Sirpac* y varios procesadores xml. Se utiliza especialmente para la creación y gestión de ontologías.

<http://ontobroker.semanticweb.org/rdfcrawl/help/specification.html>

—*RDFstore*: en esta ocasión nos encontramos con una *api* creada en *perl* que tiene como principal punto a favor el de soportar cualquier esquema *rdf*. Además es compatible con el sistema de base de datos *Berkeley*.

<http://rdfstore.sourceforge.net/>

—*Redland*: se trata de una biblioteca creada por **Dave Beckett** que genera una interfaz para que expresiones *rdf* puedan ser manipuladas, almacenadas y consultables. En la actualidad se encuentra en la versión beta, y proporciona un conjunto de utilidades muy diversas: desde parsers, hasta *apis* específicos para el módulo de consulta.

<http://www.redland.opensource.ac.uk/>

—*Xslt extractor for rdf*: en realidad no es una *API*, sino más bien de una plantilla *XSL* que se ejecuta a través de un procesador *XSL* con el fin de generar *rdf* a partir de documentos xml.

<http://www.openhealth.org/RDF/rdfExtractify.xsl>

2.- Editores

A pesar de que es perfectamente posible utilizar editores xml para generar ficheros *rdf* (como por ejemplo *XMLSpy*, *Clip*, *XED*, *Xml pro* o *tdtd*), podemos encontrar en el mercado varios editores pensados específicamente para *rdf*. Algunos de ellos también pueden ge-



Dublin Core Extraction Service

XSL file: <http://www.w3.org/2000/01/>

XML data:

Get Results

nerar ficheros en otros sistemas de metadatos, como *Dcmi* o *Gils*. Por supuesto, la mayoría trabaja a partir de dtids.

<http://www.xmlspy.com>

<http://xml.t2000.co.kr/product/clip.html>

<http://www.ltg.ed.ac.uk/~ht/xed.html>

<http://www.vervet.com/prod.html>

<ftp://ftp.mulberrytech.com/pub/tdtd>

—*DC Dot*: se trata, junto a *Reggie*, de uno de los editores más utilizados. Este servicio recupera una página web y, automáticamente, genera el conjunto de etiquetas *Dcmi* que describen su contenido. Además es posible trabajar con otros sistemas de metadatos, como *usmarc*, *Soif*, *Iafa*, *TEI*, *Gils* o *rdf*. En la última versión no sólo trabaja con páginas html, ya que también es capaz de gestionar los metadatos de documentos *Word* o *PowerPoint*.

<http://www.ukoln.ac.uk/metadata/dcdot/>

—*Gramtor*. En esta ocasión nos encontramos con un editor diferente, puesto que las relaciones se establecen de forma gráfica. La representación gráfica puede ser una serialización de xml. Como es lógico, el programa no genera un solo fichero *rdf*. También produce el fichero de la notación triple y de la posición del documento dentro del árbol.

<http://nestroy.wi-inf.uni-essen.de/xwmf/>

—*Metabrowser*: es uno de los editores que más éxito ha tenido. A su interfaz amigable se le une la posibilidad de almacenar información en *rdf*, *Dublin Core* o *Gils*. Divide el área de trabajo en tres zonas. En la primera muestra el recurso que se quiere describir. A su lado, en un árbol, aparecen las relaciones presentes dentro del objeto.

Debajo nos encontramos con el editor propiamente dicho, y es la zona donde se deben incluir las



descripciones correspondientes a cada una de las etiquetas.

<http://metabrowser.spirit.net.au>

—*Protégé*. Más que un editor de *rdf* se trata de todo un programa dedicado a la gestión del conocimiento. Se enmarca dentro de un proyecto de la *Universidad de Stanford* que pretende crear un conjunto de herramientas que utilizan metadatos para trabajar con bases de conocimiento.

<http://www.smi.stanford.edu/projects/protege/>

—*Reggie*: esta plantilla es capaz de generar descripciones basándose en la versión 3.2 ó 4.0 de html. El formato es tanto *Dublin Core* como *rdf*. Funciona de forma parecida a *DC Dot*, dado que recupera un documento web y le añade automáticamente aquellos elementos susceptibles de ser incluidos en una de las etiquetas del esquema especificado.

También es posible añadir información de forma manual, seleccionando las diversas opciones que se ofrecen dentro del applet en *java* dentro del cual se está ejecutando.

<http://metadata.net/dstc/>

—*RDFpic*. Ha sido diseñado en *java* para facilitar la incorporación de etiquetas *rdf* a la descripción de imágenes. Como esquema

rdf utiliza *Dublin Core*, ya que este sistema de metadatos aporta un conjunto de etiquetas extra que facilitan la localización y posterior recuperación de la imagen digitalizada.

<http://jigsaw.w3.org/rdfpic>

—*Rdf schema editor*. Es un prototipo experimental desarrollado en *perl* que sirve para editar, visualizar y navegar a través de documentos *rdf* basados en el esquema *rdf*. El proyecto a partir del que se creó, *Warf*, finalizó hace tiempo y el programa se ha quedado en prototipo.

http://jonas.liljegren.org/perl/proj/rdf/schema_editor/

—*S-link-s editor/publisher*. Es una aplicación *java* que se utiliza para generar documentos *rdf* que utilicen el sistema de enlace denominado *S-link-s hyperlinking*.

<http://www.openly.com/SLinkS/Editor.html>

3.- Parsers

Al igual que sucede con xml o sgml, para poder interpretar un documento *rdf* es necesaria la utilización de un procesador concreto. De la misma forma que los navegadores web hacen las funciones de procesadores html, todos los lenguajes de etiquetado necesitan una aplicación que se encargue de pro-

cesar la información contenida dentro de ellos.

Los procesadores *rdf* funcionan de forma idéntica a los de *xml*: se lee el documento, se interpreta la etiqueta correspondiente y se muestra en una interfaz de visualización más o menos amigable. Es decir, se encarga de hacer posible la presentación y distribución de documentos *rdf/xml*.

Ésta es la forma de funcionar de todos los procesadores. El resultado final, la visualización, dependerá del tipo de documento con el que se esté trabajando. Por ejemplo, si se interpreta un texto del tipo *OSD* (*open software description*), el procesador tendrá que conectarse a algún lugar remoto y capturar e instalar módulos determinados que permitirán la correcta visualización del documento (de un modo muy similar al trabajo de los plugins en los navegadores web). Si por el contrario es un documento *Pgml* (*precision graphics markup language*), el procesador hará una llamada a la interfaz gráfica ordenándole que muestre sólo una imagen. En *rdf* el procesador mostrará el texto en forma arbórea, con todos los elementos en las ramas y el contenido formando parte de las hojas.

Un procesador está compuesto, básicamente, por dos módulos: el analizador sintáctico (o parser) y la interfaz gráfica (el visualizador). Los parsers, que es la parte que nos interesa en este momento, pueden ser validadores (es decir, comprueban el documento *rdf* con la dtd correspondiente, si el documento no respeta alguna de las reglas especificadas en la dtd aparecerá un mensaje de error y el proceso de lectura del documento se detendrá) o no validadores (permiten visualizar el documento a pesar de no estar correctamente escrito, lo cual aporta una mayor rapidez en la consulta y un mayor número de errores finales).

Por ejemplo, los navegadores web incluyen analizadores sintácticos que no validan documentos. Por ese motivo son capaces de mostrar ficheros html que no están bien escritos (con etiquetas que no existen, etiquetas mal escritas o sin señalización de final de marca [/]).

La mayoría de parsers que podemos encontrar dentro del mundo *rdf* son programas sueltos que se pueden integrar en cualquier sistema fácilmente como si fuesen un módulo más. A continuación mostramos una lista con los nombres de los programas más utilizados en este sentido junto a la url donde lo-

calizarlos. La totalidad de ellos son de libre distribución y sólo es necesario adquirir una licencia de uso si se piensa en integrarlos dentro de una aplicación *rdf/xml*.

—*Datx*.

<http://www.megginson.com/DATA/X/>

—*Libwww*.

<http://www.w3.org/Library/>

—*PerlXmlParser*.

<http://www.w3.org/1999/02/26-modules/>

—*Rdf filter*.

<http://www.megginson.com/Software/rdf/rdfilter-1.0alpha.zip>

—*Rdf Graph, Rdf parser y Promenade rdf parser*.

<http://www.pro-solutions.com/download/>

—*Rdf parser in XSLT*.

<http://www.w3.org/XML/2000/04-rdf-parse/>

—*Sirpac (Simple rdf parser and compiler)*.

<http://http://www.w3.org/RDF/Implementations/SiRPAC/>

—*SWI-Prolog parser*.

<http://swi.psy.uva.nl/projects/SWI-Prolog/packages/sgml/online.html>

—*Xslt rdf parser*.

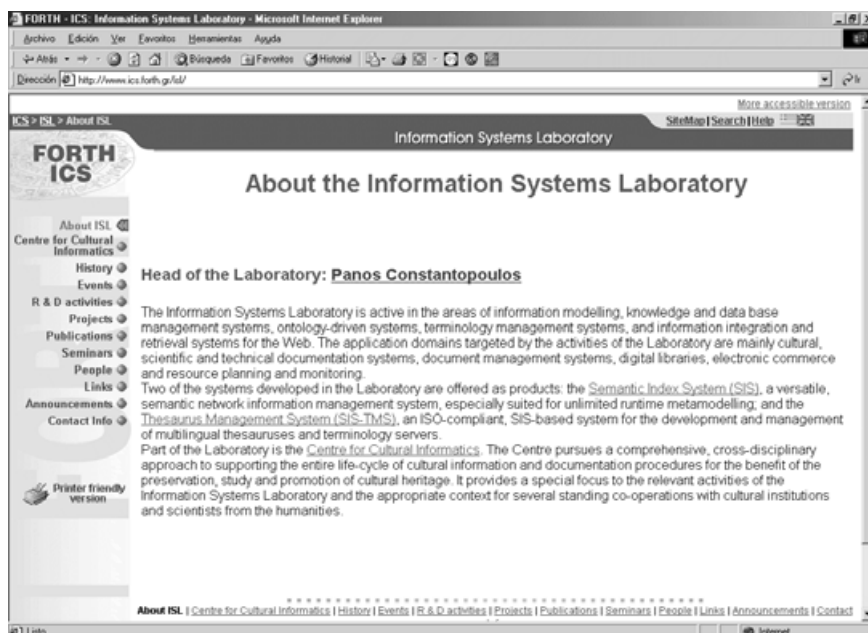
<http://http://injektilo.org/rdf/rdf.xsl>

—*Xwmlf (eXtensible web modeling framework)*.

<http://nestroy.wi-inf.uni-essen.de/xwmlf/xwmlf.html>

Junto a éstos hallamos uno que destaca especialmente: *VRP*. A pesar de encontrarse en fase de pruebas, se trata de uno de los mejores validadores del mercado. Este programa, escrito en *java*, es producto del estudio de **Tolle**, autor de uno de los trabajos más completos que existen hasta la fecha sobre el proceso de validación de documentos *rdf*.

<http://www.ics.forth.gr/proj/isst/RDF/>



4.- Interfaces con bases de datos

Lo más destacable de algunas de las direcciones que se comentarán a continuación es que ponen a disposición pública herramientas para crear y gestionar metadatos. En la mayoría de los casos lo hacen en forma de plantillas o pequeños *applets* que permiten introducir fácilmente metadatos como sistema de descripción de recursos web.

—*Algae*: es un sistema de consulta creado por el *Consortio W3* para realizar búsquedas sobre la base de datos *Algernon*. Como es habitual en el *W3C*, todo el software, realizado en *perl*, es de dominio público.

<http://www.w3.org/1999/02/26-modules/User/Algae-HOWTO.html>

—*Ginf (Generic interoperability framework)*: este modelo para implementar bibliotecas digitales está desarrollado por la *Universidad de Stanford*. Utiliza *rdf* como lenguaje de modelado de los recursos que integran la biblioteca digital. Junto a este modelo se ha implementado un programa que se encarga de convertir e intercambiar datos entre aplicaciones heterogéneas.

<http://www.diglib.stanford.edu/diglib/ginf/>

—*rdfDB*: base de datos que almacena ficheros *rdf* y que emplea como sistema de consulta un lenguaje propio basado en *SQL*. Existe también una versión *java* de la interfaz de consulta.

<http://www.guha.com/rdfdb/>

—*Wraf (Web Resource Application Framework)*: utiliza *rdf* no sólo como sistema de descripción de recursos, además la interfaz de usuario utiliza *rdf* para definir perfiles de usuario. Junto a esto, la api en *perl* que se encarga de ejecutar la aplicación tiene funciones definidas como si fueran literales de *rdf*. Dentro de este proyecto se ha

desarrollado también un editor de esquemas *rdf*.

<http://wraf.org/>

5.- Recursos online y demostraciones

—*Dublin Core extraction*: es un servicio que, automáticamente, captura un documento html o xml y genera su correspondiente codificación siguiendo el formato *Dublin Core* o *rdf*.

<http://www.w3.org/2000/06/dc-extract/form>

—*Frodo RDFsviz*: herramienta que suministra una visualización de ontologías representadas siguiendo el esquema *rdf*.

<http://www.dfki.uni-kl.de/frodo/RDFSViz/>

—*Rdf Schema Explorer*: demostración online creada por **Wolfram Conen** y **Reinhold Klapsing** basada en el parser *Rdf SWI-Prolog*. No sólo se encarga del proceso de validación, sino que además facilita la creación de esquemas *rdf*.

<http://wonkituck.wi-inf.uni-essen.de/rdfs.html>

—*The W3C rdf Xslt transformer*: es otra demostración online que utiliza uno de los parsers creados por el *W3C (O4rdf)* para, por medio de una hoja de transformación *Xslt*, generar descripciones *rdf*.

<http://www.w3.org/XML/2000/04rdf-parse/>

6.- Motores inferenciales y sistemas lógicos

—*Euler proof mechanism*: utiliza datos extraídos de páginas web para generar programas lógicos.

<ftp://windsor.agfa.be/outgoing/RC/El/NET/euler/index.html>

—*Metalog*: es un sistema de consulta compuesto por tres capas. La primera está basada en el modelo de datos de *rdf* y proporciona el sistema para expresar relaciones lógicas. Con estas relaciones se construyen las reglas de inferencia

que serán utilizadas más adelante para las búsquedas. La segunda capa es una interpretación en lógica formal del modelo de datos *rdf*. Por último, la tercera capa está formada por la interfaz para escribir los datos estructurados y las reglas de inferencia.

<http://www.w3.org/RDF/Metalog>

—*OilEd*: es un editor de ontologías que genera nuevas ontologías que utilicen el lenguaje *OIL (Ontology inferencel/interchange layer)*.

<http://img.cs.man.ac.uk/oil>

—*Protégé*: herramienta que permite construir una ontología de un dominio determinado. Permite también personalizar la forma en la que el sistema adquirirá conocimiento futuro. Todo esto lo hace utilizando *rdf* como lenguaje para expresar relaciones entre objetos de conocimiento.

<http://www.smi.stanford.edu/projects/protége/whatis.html>

—*Silri (Rdf interpreter)*: es el módulo principal de un programa realizado en *java* y que transforma el modelo de datos *rdf* en reglas lógicas para, después, realizar consultas sobre ellas. Utiliza también el parser *Sirpac* para comprobar que el documento fuente no genera problemas de lógica.

http://www.ontoprise.de/start_dwnlo.htm

—*SIS (Semantic Index System)*: utilidad para describir y documentar documentos y las relaciones que estos tienen entre sí.

<http://www.ics.forth.gr/proj/isst>

Bibliografía

Brickley, Dan y **Guha, R. V.** *Resource Description Framework (rdf) schema specification*. <http://www.w3.org/TR/PR-rdf-schema>

Hjelm, Johan. *Creating the semantic web in rdf*. New York: Wiley Computer Publishing; John Wiley & sons, Inc.; 2001.

Iannella, Renato. *An idiot's guide to the Resource Description Framework*. 1998. <http://archive.dstc.edu.au/RDU/reports/RDF-Idiot/>

Lassila, Ora. *Introduction to rdf metadata.*
1997.
<http://www.w3.org/TR/NOTE-rdf-simple-intro>

Miller, Eric. "An introduction to the Resource Description Framework". En: *DLib magazine*.
Mayo 1998.
<http://www.dlib.org/dlib/may98/miller/05miller.html>

Tolle, Karsten. *Validating rdf parser (VRP) - analyzing and parsing rdf.* Keraklion; 2000;
Technical report forth-ics-tr-270.
<http://139.91.183.30:9090/RDF/>

Jose A. Senso
jsenso@ugr.es

El profesional de la información está abierto a todos los bibliotecarios, documentalistas y otros profesionales de la información, así como a las empresas y organizaciones del sector para que puedan exponer sus noticias, productos, servicios, experiencias y opiniones.

Dirigir todas las colaboraciones para publicar a:

El profesional de la información

Apartado 32.280

08080 Barcelona

Fax: +34-932 701 145

epi@sarenet.es