

PROBLEMÁTICA Y TENDENCIAS EN LA ARQUITECTURA DE METADATOS WEB

*Pedro Manuel Díaz Ortuño**

Facultad de Ciencias de la Documentación. Universidad de Murcia.

Resumen: El propósito de este informe es poner de manifiesto las dificultades que entraña el despliegue efectivo del Web Semántico e identificar los acuerdos y realizaciones alcanzadas. Indicaremos las dificultades relacionadas con la recuperación de información en el Web "sintáctico" actual y la propuesta del W3C como solución a estos problemas. Comentaremos las recomendaciones y propuestas de lenguajes conceptuales, valorando su aportación al Web semántico, y las principales discrepancias y cuestiones abiertas en el desarrollo de las especificaciones. Se considerarán otras tecnologías e infraestructuras, los perfiles de aplicación y los registros de esquemas, importantes para la creación y utilización de registros de metadatos en diferentes comunidades, el descubrimiento de recursos y el establecimiento de correspondencias entre distintos formatos que permiten la interoperabilidad entre aplicaciones.

Palabras clave: Metadatos; esquemas de metadatos; modelos de datos; web semántico; RDF.

Title: ISSUES AND TRENDS IN WEB METADATA ARCHITECTURE

Abstract: The purpose of this report is to show the difficulties that the effective unfolding of the Semantic Web involve and to identify the achieved agreements and realizations. The difficulties related to information retrieval in the current "syntactic" Web and the proposal of the W3C as a solution of these problems will be indicated. The recommendations and proposals of conceptual languages, valuing their contribution to the semantic Web, and the main discrepancies and questions opened up in the development of the specifications will be discussed. Other technologies and infrastructures, i.e. the application profiles and the schemes registries, important for the creation and use of metadata records in different communities, the discovery of resources and the establishment of correspondences among different formats that allow the interoperability among applications will be commented.

Keywords: Metadata; metadata schemes; data models; semantic web; RDF.

1. INTRODUCCIÓN

Los motores de búsqueda Web, aunque han mejorado mucho en los últimos años, continúan catalogando sólo una porción pequeña de la Web, producen resultados que a veces son no pertinentes, a menudo inexactos, o imposibles de encontrar (enlaces rotos). Están limitados por varios factores: la tecnología usada, la cantidad y calidad de metadatos disponibles, y el tamaño de la Web.

Podemos clasificar a los motores de búsqueda en dos categorías: robots y directorios. La diferencia principal entre los dos es el método de recolección de datos.

* diazor@um.es

Un robot es un programa que rastrea la estructura hipertextual de la Web, recogiendo información sobre las páginas que encuentra. Esa información se indiza y se introduce en una base de datos que será explorada posteriormente utilizando un motor de búsqueda.

Los motores de búsqueda basados en directorios, como Yahoo! y Dmoz¹, usan catálogos creados manualmente. Estos catálogos contienen referencias a sitios Web descubiertas por el personal del directorio o remitidas por los administradores de sitios web. Las referencias contenidas en un directorio generalmente son clasificadas por personal altamente especializado, implicando un gran consumo de tiempo. Tienen un mayor grado de exactitud y pertinencia (precisión) que los motores basados en robots pero producen un conjunto de resultados de búsqueda mucho más pequeño (menor exhaustividad) debido al esfuerzo manual involucrado y al tamaño de Internet. El aumento continuo de documentos existentes en la Web y su inestabilidad ocasionan que los directorios clasificados por humanos sólo sean capaces de cubrir una parte de la red y que sean difíciles de actualizar.

La Web fue desarrollada originalmente como un producto consumible por humanos. Si bien todo en el Web es "legible" para las máquinas, esta información no es "entendible" para las mismas. La *Semantic Web Activity*² es un proyecto desarrollado por el Consorcio Web (*World Wide Web Consortium*, W3C). Su objetivo es llevar significado, estructura, y organización al Web de modo que posibilite el descubrimiento automatizado, la comprensión, y el intercambio de recursos Internet.

El W3C plantea la necesidad de crear una "capa semántica" basada en sistemas de metadatos "entendibles" por las máquinas y que sirva para describir el Web. Presenta el web semántico como una extensión del Web actual en la cual se da un significado bien definido a la información, permitiendo un mejor trabajo en colaboración, tanto a computadoras como a personas.

RDF (Resource Description Framework) es una de las tecnologías claves en el proyecto de la Semantic Web Activity. RDF es la propuesta del W3C para definir metadatos en el Web y la base para el procesamiento de metadatos: proporciona interoperabilidad semántica entre aplicaciones que intercambian información entendible por máquina. RDF es simplemente un modelo de datos que permite crear metadatos legibles y entendibles por máquina. La interoperabilidad semántica de sistemas de metadatos implica significados compartidos y gramáticas compartidas. Como con el lenguaje natural, traducir un sistema de los metadatos particular en los términos y gramática de otro requiere interpretación y puede involucrar pérdida o distorsión de significado. El reconocimiento y la aceptación de este límite inherente a la interoperabilidad es una marca constante en la filosofía del Web Semántico³.

El web semántico y RDF abordan estos problemas proporcionando una infraestructura que posibilite la automatización del descubrimiento de recursos (como los motores basados en robots) y la capacidad para indicar los recursos inteligentemente (como los motores

¹ Dmoz, <http://www.dmoz.org>. La base de datos de Dmoz es pública y puede ser descargada. Utiliza XML para almacenar sus datos utilizando un vocabulario RDF para describir la estructura de Dmoz. Básicamente puede accederse a dos archivos, uno con la estructura del árbol de categorías (sin links) y otro con todas las descripciones y sus correspondientes links.

² W3C Semantic Web Activity, <http://www.w3.org/2001/sw/>

³ Proyecto SCHEMAS. *The SCHEMAS Forum – a Retrospective Glossary*. UK Office for Library and Information Networking (UKOLN), Universidad de Bath, actualización de 6 de marzo de 2002. Disponible en: <http://www.schemas-forum.org/info-services/d74.htm>

basados en directorios). RDF lo logra a través del uso de semántica entendible por máquina. Se diseñan los metadatos en RDF específicamente para ser entendidos e intercambiados por procesos automatizados, como los agentes software y los sistemas de búsqueda. RDF intentará mejorar el descubrimiento de recursos proporcionando un mayor grado de precisión en los resultados de las búsquedas que los sistemas de búsqueda actuales. Por ejemplo, una solicitud RDF para “todos los recursos escritos por Ora Lassila en 2002” produciría un conjunto de resultados conciso y completamente pertinente.

2. PRINCIPIOS DE METADATOS

Duval, Hodgins, Sutton y Weibel manifiestan en un reciente artículo los principios de acuerdo compartidos por dos importantes iniciativas de metadatos⁴: la *Dublin Core Metadata Initiative* (DCMI) y el *Institute for Electrical and Electronics Engineers* (IEEE) *Learning Object Metadata Working Group* (LOM). Este acuerdo surgió de una reunión conjunta del grupo de trabajo (taskforce) de metadatos en Ottawa en agosto de 2001. Los principios son los conceptos considerados comunes a todos los dominios de metadatos y que podrían servir de base en el diseño de cualquier *esquema de metadatos* o aplicación.

Modularidad

Debe ser alcanzable una flexibilidad en la arquitectura de metadatos Web, que permita a los diseñadores **mezclar una variedad de módulos semánticos en un esquema compuesto**, dentro de un fundamento sintáctico común (como XML). En una arquitectura modular de metadatos, los elementos de datos de esquemas diferentes así como los vocabularios y otros elementos constructivos pueden combinarse de una manera interoperable sintáctica y semánticamente. Pueden ensamblarse conjuntos modulares basados en esquemas de metadatos preestablecidos que incluyan la funcionalidad de cada componente y que reúnan los requisitos específicos de una determinada aplicación.

Extensibilidad

Las arquitecturas de metadatos deben acomodar la **noción de un esquema base con elementos adicionales** que ajusten una aplicación dada a las necesidades locales o las necesidades específicas del dominio sin comprometer excesivamente la interoperabilidad proporcionada por el esquema base. Los sistemas de los metadatos deben permitir **extensiones** para que puedan acomodarse a las necesidades particulares de una determinada aplicación. Algunos elementos probablemente serán comunes a la mayoría de los esquemas de metadatos (por ejemplo, el concepto de creador o identificador de un recurso de información) mientras que otros serán específicos a aplicaciones particulares o dominios.

Refinamiento

Los dominios de aplicación diferirán de acuerdo con el nivel de detalle necesario o deseable. El diseño de estándares de metadatos debería permitir a los diseñadores de esquemas el nivel de detalle apropiado a la aplicación. Poblar con metadatos las bases de datos es costoso, por lo que hay fuertes incentivos económicos para crear metadatos con el sufi-

⁴ Duval, E.; Hodgins, W.; Sutton, S.; Weibel, S. *Metadata Principles and Practicalities*. D-Lib Magazine, v.8, n.4, abril de 2002. Disponible en : <http://www.dlib.org/dlib/april02/weibel/04weibel.html>

ciente nivel de detalle para cumplir los requisitos funcionales de una aplicación, pero no más.

Se consideran varios tipos de refinamientos. La adición de **calificadores** refinan o hacen más específico el significado de un elemento: ilustrador, autor, compositor, o escultor son ejemplos de tipos particulares de un término más general, creador.

Una segunda variedad de refinamiento involucra la especificación de esquemas particulares o conjuntos de valores para un determinado elemento. Contando con un conjunto de valores común puede incrementarse la interoperabilidad semántica entre aplicaciones.

El uso de vocabularios controlados es otra aproximación importante que mejora la precisión de las descripciones y permite aprovechar la inversión intelectual realizada por muchos dominios en la mejora del acceso por materias a los recursos. Hay centenares de tesauros específicos de dominios y sistemas de la clasificación que pueden importarse a la arquitectura de metadatos Web y considerable el esfuerzo investigador en este ámbito. Cross y Brickley presentan una propuesta de implementación de tesauros y sistemas de clasificación con RDF. En su propuesta, definen el vocabulario básico RDF, el esquema en RDFS y proporcionan un ejemplo de tesoro en RDF⁵.

Plurilingüismo

Es esencial adoptar arquitecturas de metadatos que respeten la diversidad lingüística y cultural. La Web alcanzará su potencial como sistema de información global, si los recursos están disponibles a los usuarios en sus idiomas nativos, en los conjuntos de caracteres apropiados, y con los metadatos apropiados a la gestión de los recursos.

Los estándares tratan estos problemas a través de **procesos complementarios de internacionalización** (creación de estándares neutrales) y **localización** (adaptación de un estándar neutral a un contexto local).

3. TECNOLOGÍAS E INFRAESTRUCTURA EN EL WEB SEMÁNTICO

La puesta en práctica de estos principios de diseño de conjuntos de metadatos implica el desarrollo de una infraestructura habilitadora. Consideraremos algunos aspectos importantes de la emergente tecnología de creación y gestión de metadatos en Internet.

3.1.Namespaces

La noción de *namespaces* (espacios de nombres) es una parte fundamental de la infraestructura Web (y particularmente en XML⁶), aunque el concepto es anterior a la Web y resulta familiar a la mayoría. Son una parte crítica de la infraestructura necesaria para desplegar **sistemas de metadatos modulares** en el Web.

XML fue creado para permitir la interoperabilidad. Puesto que todo el mundo puede crear sus propios vocabularios XML, se produciría una terrible confusión si diferentes desarrolladores optasen por los mismos nombres de elementos para representar diferentes

⁵ Cross, P.; Brickley, D. *Conceptual relationships for encoding thesauri, classification systems and organised metadata collections and a proposal for encoding a core set of thesaurus relationships using an RDF Schema*. Institute for Learning & Research Technology, University of Bristol, UK., 6 de junio de 2002. Disponible en: <http://www.desire.org/results/discovery/rdftheschema.html>

⁶ W3C World Wide Web Consortium. *Namespaces in XML*. Tim Bray, Dave Hollander, y Andrew Layman (eds.). 14 de enero de 1999. Disponible en: <http://www.w3.org/TR/REC-xml-names>

entidades. Los “espacios de nombres” (*NameSpaces*) fueron introducidos en XML para resolver conflictos de nombres entre elementos en un documento XML cuando los elementos se derivan de diferentes fuentes, permitiendo el uso de múltiples vocabularios en un mismo documento. Un namespace es un vocabulario definido dentro de un URI (*Universal Resource Identifier, Identificador de Recursos Universal*).

Los namespaces posibilitan alcanzar los principios de modularidad y extensibilidad. Cualquier conjunto de elementos de metadatos es un *namespace* limitado por las reglas y convenciones determinadas por su agencia de mantenimiento. Las declaraciones de namespace permiten al diseñador de esquemas de metadatos definir el contexto para un término particular, asegurando que el término tiene una única definición dentro de los límites del namespace declarado. Usando esta infraestructura, los diseñadores del sistema de metadatos pueden seleccionar los elementos del conjunto de elementos de metadatos existentes convenientemente establecidos, evitando **reinventar** los conjuntos de los metadatos establecidos para cada nuevo dominio.

3.2.Sintaxis y Semántica

Para que dos comunidades puedan compartir metadatos es necesario llegar a acuerdos tanto en la forma como en su significado. Dos comunidades pueden estar de acuerdo sobre el significado de un elemento de metadatos, pero hasta que no tengan una convención compartida para identificar y codificar los valores, no podrán intercambiar metadatos fácilmente.

Hemos conocido varias versiones de HTML, la aparición de XML y el desarrollo de tecnologías derivadas entre las que podemos incluir los Esquemas XML y los Esquemas RDF. La falta de estabilidad subraya la necesidad de intentar mantener la independencia entre la semántica de los elementos de metadatos y su representación sintáctica.

RDF (Resource Description Framework)

Básicamente, RDF proporciona un modelo de datos para describir la semántica de los datos de una forma procesable por máquina. Un objeto de información o recurso se describe a través de un conjunto de propiedades. La esencia de RDF es pues un modelo formal para la representación de las propiedades y los valores de esas propiedades⁷. RDF usa los namespaces XML para calificar los recursos. Esto previene colisiones de nombre de elemento que podrían suscitarse cuando se combinan múltiples vocabularios. Los namespaces también se usan para identificar el esquema RDF.

El modelo de datos básico consiste en tres tipos de objetos:

- **Recursos:** Todo aquello que puede ser descrito por una expresión RDF es un "recurso". Puede ser una página web, o una parte de una página web, una colección entera de páginas o un sitio Web. Un recurso también puede ser un objeto que no es directamente accesible vía web como por ejemplo un libro. Todos los recursos son identificados mediante URIs (entendidos como identificadores de objetos). Es importante destacar que cualquier cosa puede tener un URI y su extensibilidad permite la **definición de identificadores** para cualquier objeto imaginable. Un URI

⁷ W3C World Wide Web Consortium. *Resource Description Framework (RDF): Model and Syntax Specification. W3C Recommendation, 22 February 1999*. [Ora Lassila y Ralph R. Swich, eds. Disponible en: <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>

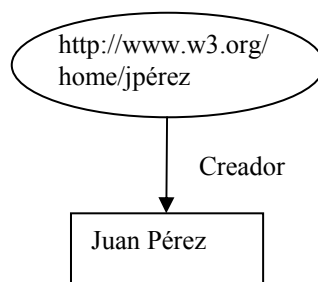
no necesariamente direcciona un recurso Web. Por ejemplo, el ISBN 0679405739, puede ser usado como un URI "ISBN:0679405739".

- **Propiedades:** Una propiedad es una característica, aspecto, atributo o relación usada para describir un recurso. Cada propiedad tiene un significado específico, define sus valores posibles, los tipos de recursos que puede describir y su relación con otras propiedades.
- **Sentencias:** Un recurso junto con una propiedad con nombre y con el valor de la propiedad para dicho recurso se denomina una "sentencia". Las sentencias representan relaciones binarias específicas entre dos objetos. Las tres partes de una sentencia se denominan respectivamente sujeto (recurso), predicado (propiedad) y objeto (recurso o literal). El **objeto** (el valor de la propiedad) puede ser **otro recurso** (especificado por una URI) o bien **un valor literal**.

Consideremos la siguiente oración: "*Juan Pérez es el creador del recurso <http://www.w3.org/home/jperez>*"

La oración tiene las siguientes partes:

Sujeto (recurso)	http://www.w3.org/home/jperez
Predicado (propiedad)	Creador
Objeto (literal)	Juan Pérez



El modelo RDF puede ser representado como un **grafo dirigido etiquetado**. En estos grafos, los nodos (representados como elipses) representan recursos y los arcos representan propiedades con nombre (el rótulo del arco es el nombre de la propiedad). Los nodos literales se dibujan como rectángulos.

Como ejemplo de un editor RDF con representación de grafos podemos destacar a **IsaViz**, un entorno para la visualización y edición de modelos RDF. Recursos y literales son los nodos del grafo (elipses y rectángulos), con propiedades representadas como arcos que conectan estos nodos⁸.

⁸ W3C IsaViz, <http://www.w3.org/2001/11/IsaViz/>

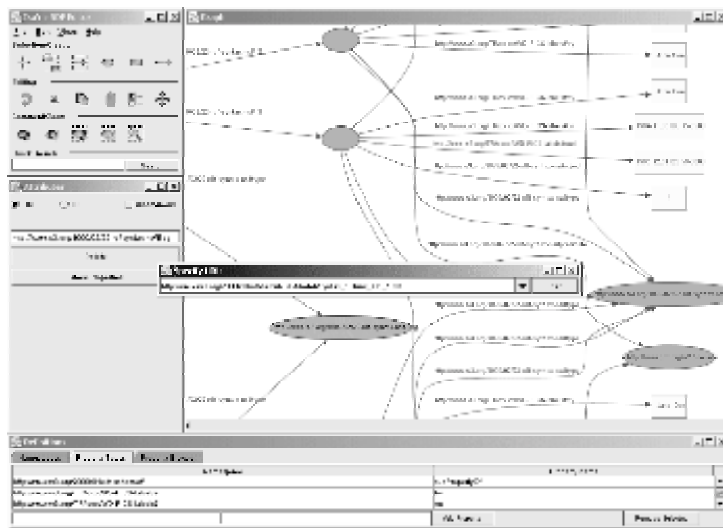
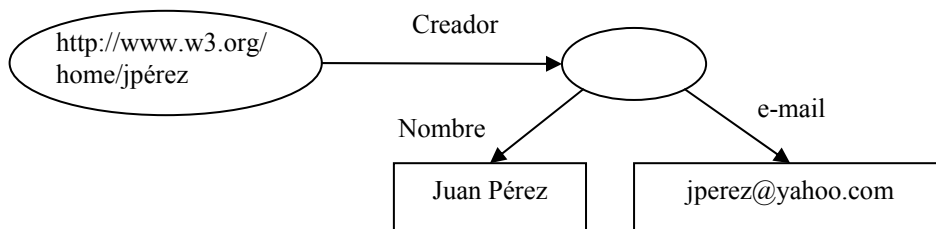


Fig. 1. Visualización de un grafo RDF en Isaviz

Supongamos que queremos decir algo más sobre las características del creador del recurso, por ejemplo: “El individuo cuyo nombre es Juan Pérez, email jperez@yahoo.com es el creador de <http://www.w3.org/home/jperez>”

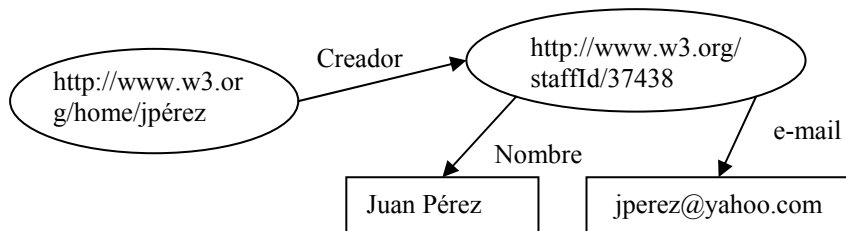


La idea de esta sentencia es destacar que el valor de la propiedad Creador es una entidad estructurada. Para permitir sentencias sobre sentencias, una sentencia puede ser reificada (*Reificación*⁹, sentencia expresada como un recurso con un conjunto de propiedades). El objeto es el contenedor de la sentencia original (sentencia reificada) y puede ser usado para formar nuevas sentencias acerca de la sentencia original.

En RDF este tipo de entidades se representan como nuevos recursos. La sentencia anterior no le da un nombre al recurso, es *anónimo*, por lo que en el diagrama se lo representa como un ovalo vacío.

La entidad del ejemplo anterior puede también recibir un identificador único, la designación del identificador es responsabilidad del diseñador de la aplicación. Por ejemplo supongamos que se usa un identificador de empleado como identificador de los recursos tipo persona. Y supongamos que la URI es de la forma: <http://www.w3.org/staffId/37438>. El siguiente grafo refleja esta construcción.

⁹ DRAE, *Cosificación*: acción y efecto de cosificar. *Cosificar*: Convertir algo en cosa.; considerar como cosa algo que no lo es, por ejemplo, una persona.



Sintaxis RDF básica

RDF sirve para describir la semántica. Sin embargo la semántica, en un sistema informático, sirve de poco sin una sintaxis asociada. RDF usa una serialización XML conocida como RDF/XML que proporciona la base sintáctica¹⁰. Se han desarrollado también varios lenguajes de consulta para RDF que soportan esta sintaxis.

Una sentencia RDF raramente aparece en forma aislada. Lo normal es que varias propiedades de un recurso sean indicadas simultáneamente. La sintaxis RDF/XML ha sido diseñada para permitir agrupar varias sentencias sobre un mismo recurso en un elemento "Description". El elemento "Description" menciona en un atributo "about" el recurso al que se aplican las sentencias. Si el recurso todavía no existe, el elemento "Description" puede asignarle un identificador en el momento usando un atributo ID.

Tomando como base el ejemplo anterior, se escribe en RDF/XML de la forma:

```

<rdf:RDF>
  <rdf:Description about="http://www.w3.org/home/jperez">
    <s:creator>Juan Pérez</s:creator>
  </rdf:Description>
</rdf:RDF>
  
```

El segundo ejemplo “*El individuo cuyo nombre es Juan Pérez, email jperez@yahoo.com es el creador de http://www.w3.org/home/jperez*” podría escribirse en RDF/XML de la forma:

```

<rdf:RDF>
  <rdf:Description about="http://www.w3.org/home/jperez">
    <s:creator rdf:resource="http://www.w3.org/staffId/37438" />
  </rdf:Description>
  <rdf:Description about="http://www.w3.org/staffId/37438">
    <v:name>Juan Perez</v:name>
    <v:email>jperez@yahoo.com</v:email>
  </rdf:Description>
</rdf:RDF>
  
```

Frecuentemente es necesario referirse a una colección de recursos. RDF define tipos específicos de **contenedor** que representan multiconjuntos, secuencias, y alternativas. RDF define contenedores para manejar listas de recursos o literales:

- **Bag:** Un bag es una lista de recursos o literales sin orden. Se permiten valores duplicados y no hay diferencias en cuanto al orden en que aparecen los recursos.

¹⁰ W3C World Wide Web Consortium. *RDF/XML Syntax Specification (Revised)*. W3C Working Draft 25 March 2002. Dave Beckett, ed. W3C, 25 de marzo de 2002. Disponible en: <http://www.w3.org/TR/2002/WD-rdf-syntax-grammar-20020325>

- **Sequence:** Una secuencia es una lista ordenada de recursos o literales. Se permiten valores duplicados.
- **Alternative:** Es una secuencia de recursos o literales para un "único" valor o propiedad. De la lista de recursos o literales debe elegirse uno.

RDF Schema

Las primitivas de modelado de datos proporcionadas por RDF son muy básicas (identificación de objeto, relaciones binarias, reificación y contenedores). La *RDF Schema Specification*¹¹ es una extensión de RDF que proporciona primitivas adicionales. Enriquece el modelo básico, proporcionando un vocabulario para RDF, que se asume tiene una cierta semántica. Permite a los diseñadores especificar una jerarquía explícita de clases de recursos y propiedades que describen estas clases, junto con las restricciones sobre las combinaciones permitidas de clases, propiedades y valores.

Comentamos en detalle algunas características de RDFS:

- **Clases:** *rdfs:Resource*, *rdf:Property*, y *rdfs:Class*.
Cualquier cosa descrita por una sentencia RDF se considera una instancia de la clase *rdfs:Resource*. La clase *rdf:Property* es la clase de todas las propiedades utilizadas en la caracterización de las instancias de *rdfs:Resource*. Finalmente, *rdfs:Class* se usa para definir conceptos. Cada concepto debe ser una instancia de *rdfs:Class*.
- **Propiedades:** *rdf:type*, *rdfs:subClassOf*, and *rdfs:subPropertyOf*.
La relación *rdf:type* modela interrelaciones del tipo *instancia-de* entre recursos y clases. Un recurso puede ser una instancia de más de una clase. *Rdfs:subClassOf* modela la jerarquía de clases, donde una clase puede ser subclase de otras subclases. Si una propiedad P2 es una subpropiedad de (*rdfs:subPropertyOf*) otra propiedad P1, y si un recurso R tiene una propiedad P2 con valor V, esto implica que el recurso R también tiene la propiedad P1 con valor V.
- **Restricciones y clases:** *rdfs:ConstraintResource*, *rdfs:ConstraintProperty*, *rdfs:range*, y *rdfs:domain*.
rdfs:ConstraintResource define la clase de todas las restricciones. *rdfs:ConstraintProperty* es un subconjunto de *rdfs:ConstraintResource*. Tiene dos instancias: *rdfs:range* y *rdfs:domain* que se usan para restringir el rango y el dominio de las propiedades. No se permite expresar más de una restricción de rango sobre una propiedad¹². En dominios si se permite, y se interpreta como una unión de dominios.

El siguiente ejemplo se ha tomado de la especificación de RDFS 1.0.

¹¹ W3C World Wide Web Consortium. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Working Draft 30 April 2002. Dan Brickley y R. V. Guha, eds. W3C, 30 de abril de 2002. Disponible en: <http://www.w3.org/TR/2002/WD-rdf-schema-20020430/>

¹² Aunque no está recogida en la propuesta original de RDFS, podemos encontrar una recomendación para implementadores que si la permite. El grupo RDF Core la incluirá en la especificación final.

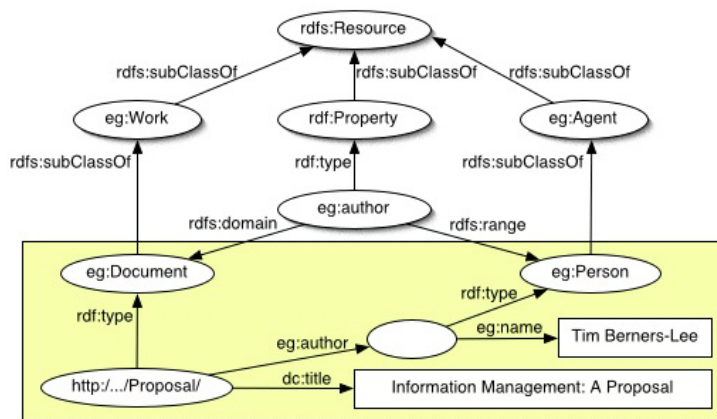


Fig. 2. Ejemplo de grafo RDFS

XML Schemas frente a RDF Schemas

Confusamente, el término “esquema” se asocia con dos especificaciones del W3C rivales, el esquema XML y el esquema RDF. En un sentido amplio, un esquema XML se diseña para analizar y **validar** la estructura de etiquetas de archivos de metadatos (en este sentido, un esquema XML es un “esquema de documento”). En contraste, para representar las relaciones de términos particulares con otros términos del esquema o con términos definidos en otros esquemas en el Web, es preferible un esquema RDF (en este sentido, un esquema RDF es un esquema semántico).

A pesar de la similitud en sus nombres, RDFS desempeña un papel diferente al de XMLS. XMLS, como las DTDs, prescribe el orden y combinación de las etiquetas en un documento XML. En contraste, RDFS proporciona información sobre la interpretación de las sentencias contenidas en un modelo de datos RDF, pero no restringe la apariencia sintáctica de una descripción RDF.

Desarrollados independientemente por dos grupos de trabajo paralelos a finales de los 90, la relación entre estas dos especificaciones es la causa de muchas confusiones, y desde enero del 2002, el W3C está liderando los esfuerzos para combinar las funcionalidades de ambos en un lenguaje de esquema integrado.

Patel-Schneider y Siméon¹³ señalan las diferencias entre RDFS y XMLS: XML es ordenado, RDF no; XML usa un modelo de árbol, RDF de grafos; RDF distingue entre clases y propiedades, y en XML, todos son elementos. Sin embargo, cree que es posible desarrollar un modelo unificado que sirva como base para las aplicaciones que trabajen tanto con documentos (XML) como con semántica (RDF). El modelo Ying/Yang se desarrolla sobre el modelo de datos de Xquery 1.0 y de Xpath 2.0. Su aproximación se basa en la incorporación de extensiones de tipos de datos XMLS en documentos XML/RDF.

Su modelo no resuelve aspectos como la reificación y contenedores de RDF, ni tiene claro como incorporar las cardinalidades de XMLS a un entorno semántico. Tampoco está claro si sería más apropiado consultar el modelo Ying/Yang con un lenguaje basado en

¹³ Patel-Schneider, P.; Siméon, J. *The Ying/Yang Web: XML syntax and RDF semantics*. WWW2002, Honolulu, Hawaii, mayo de 2002. Disponible en: <http://www2002.org/CDROM/refereed/231/index.html>

extensiones de XQuery (XML), extendiendo lenguajes de consulta RDF como RQL, o diseñando un lenguaje unificado que proporcione características de ambos lenguajes.

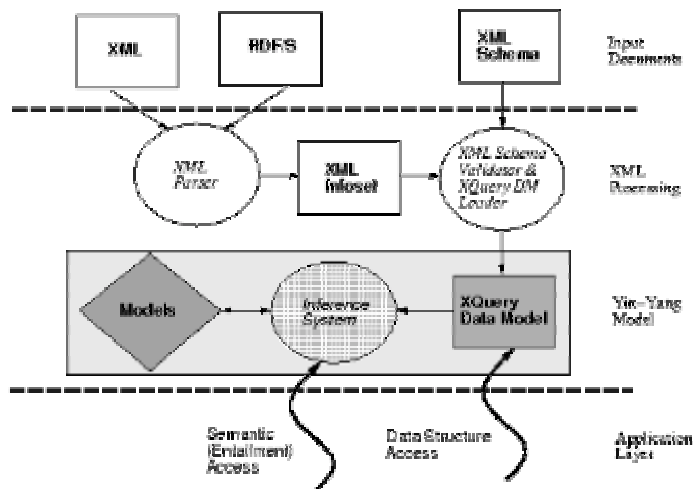


Fig. 3. Modelo Ying/Yang de Patel-Schneider y Simeon.

Tipos de datos de XMLS en RDFS

No deja de ser sorprendente que significativas iniciativas de diferentes comunidades de los metadatos (IMS-LOM, OAI, Bath Profile, ODRL, E-Book, UDDI, DOI,...) estén proporcionando sus especificaciones (namespaces y perfiles de aplicación) en XML Schema, no en RDFS. Sin embargo, DCMI continúa, tal vez nadando contra corriente, en el camino de RDF/RDFS¹⁴.

Powell y Johnston, desarrollan las directrices (como propuesta de recomendación) para la implementación de aplicaciones de metadatos Dublin Core usando XML¹⁵. El modelo de metadatos DC se describe de una forma sintácticamente neutral, junto con algunas directrices específicas para implementaciones XML. El documento no proporciona las recomendaciones para la codificación de DC en RDF/XML. Tampoco adopta ninguna postura acerca de las ventajas relativas de codificar metadatos en XML plano con respecto a una codificación RDF/XML. *Indican que las directrices se proporcionan para aquellos casos en los que no es apropiada una codificación RDF/XML*. Los mecanismos para la codificación de DC en RDF/XML están siendo desarrollados por el DCMI Architecture Working Group¹⁶.

Como veremos más adelante, los perfiles de aplicación requieren del uso de primitivas de modelado no proporcionadas por la especificación actual de RDFS. Es necesario utili-

¹⁴ *Guidance on expressing the Dublin Core within the Resource Description Framework (RDF). Draft Proposal.* Eric Millar, Paul Millar y Dan Brickley, eds. DCMI, 1 de julio de 1999. Disponible en <http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/>

¹⁵ *Guidelines for implementing Dublin Core in XML.* DCMI, Proposed Recommendation. A. Powell y P. Johnston, eds. DCMI, 23 de Julio de 2002. Disponible en: <http://www.ukoln.ac.uk/documents/2002/07/23/dc-xml-guidelines/>

¹⁶ DCMI, Architecture Working Group, <http://dublincore.org/groups/architecture/>

zar la adecuada combinación de tecnologías dentro del trabajo a desarrollar en torno a los metadatos. RDF como modelo conceptual, XML para la sintaxis, XSLT para las transformaciones, etc.

La segunda parte de la especificación de XML Schema (tipos de datos) es la parte que está atrayendo la atención de la comunidad DC. Muchos de sus miembros demandan una mayor precisión sobre las codificaciones, tipos de datos básicos, usadas en los metadatos DC. Sergey Melnik sugiere que la especificación de RDF Schema debería usar los tipos de datos de XML Schema en ejemplos o/y alguna especificación formal de las correspondencias de estos tipos de datos con el modelo RDF¹⁷. Es una cuestión que se está debatiendo en el grupo de trabajo RDF Core.

OWL (Web Ontology Language)

OWL es un lenguaje en desarrollo por el *W3C Ontology Working Group* (WebOnt¹⁸) para publicar y compartir ontologías (definición del vocabulario de términos y relaciones) en el Web. El OWL se deriva de DAML+OIL y, al igual que este lenguaje, es una extensión de RDFS.

En el contexto del Web Semántico, las ontologías suponen una capa por encima de RDF y los esquemas RDF. DAML+OIL es una propuesta de lenguaje de modelado de ontologías surgida del trabajo conjunto desarrollado por la Iniciativa DAML (DARPA's Agent Markup Language) y el OIL Consortium (OIL). Los lenguajes como OIL, DAML, DAML+OIL y OWL son cruciales para el desarrollo del Web Semántico. Sin semánticas bien definidas y procedimientos de inferencia, los agentes software no serán capaces de procesar consistentemente la información.

Las declaraciones en OWL definen las clases, propiedades e individuos (*instancias*), junto con propiedades ya definidas en RDFS (*subclase*, *subpropiedad*, *dominio*, *rango*) a las que se añaden otras como la relación inversa entre propiedades (*inverse-of*), y un conjunto de axiomas adicionales para definir restricciones, como cardinalidades de propiedades de instancias de clase (*cardinality*, *mincardinality*, *maxcardinality*), enumeraciones de instancias de clase (*uno-de*, *one-of*) y combinaciones de clases (intersecciones, uniones y complementos de clases). Incorpora una clase general denominada "*Thing*" que es la clase de todos los individuos y la superclase de todas las clases, y una clase "final" especial con el nombre de "*Nothing*" que es la clase vacía. Las propiedades se usan para describir una relación entre individuos¹⁹. Se espera que incluya definiciones de tipos de datos de XML Schema (`rdf:type` del valor y representación con `rdf:value`, ej. `<xsd:decimal rdf:value="10.5">`).

¹⁷ W3C, RDFCore Working Group, RDF Issues Tracking, <http://www.w3.org/2000/03/rdf-tracking/>

¹⁸ W3C Web Ontology Working Group (WebOnt). <http://www.w3.org/2001/sw/WebOnt/>

¹⁹ W3C. *OWL Web Ontology Language 1.0 Reference. W3C Working Draft 29 July 2002*. Mike Dean, Dan Connolly, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider y Lynn Andrea Stein, eds. W3C, .29 de julio de 2002. Disponible en: <http://www.w3.org/TR/2002/WD-owl-ref-20020729/>

OWL Lite proporciona un subconjunto de OWL, bastante sencillo y útil desde el punto de vista de los desarrolladores de herramientas²⁰. Se basa en las características de DAML+OIL normalmente usadas en la construcción de ontologías.

La sintaxis de intercambio oficial será RDF/XML aunque se ha planteado una codificación XML alternativa, junto con una representación UML²¹.

El primer borrador de la especificación se publicó a finales de julio de 2002. Se espera que se convierta en recomendación candidata a finales de 2002. En particular, se está abordando la problemática relativa a los tipos de datos en comunicación constante con el grupo de trabajo RDF Core²², siendo muy probable la integración de los tipos de datos XMLS en la recomendación final.

Editores de ontologías basados en marcos

La emergencia del Web Semántico será una realidad si se llega a una masa crítica, a una amplia comunidad Web, que acepte y despliegue las tecnologías de representación de conocimiento (KR, Knowledge Representation) distribuidas, y en este contexto la facilidad de comprensión del paradigma de KR es de gran importancia²³.

Una parte esencial en la arquitectura del Web Semántico es la capacidad de los productores de contenidos para compartir la información ontológica que define la semántica formal del contenido del Web. El sentido filosófico del término *ontología*, “la parte de la metafísica, que trata del ser en general y de sus propiedades trascendentales”, hace referencia a la esencia misma del ser, a su existencia. Para los sistemas de IA (Inteligencia Artificial), lo que “existe” es “*lo que puede representarse*”²⁴. Una ontología es una descripción formal de los conceptos y de las relaciones existentes entre esos conceptos. Las ontologías pueden usarse para proporcionar una especificación concreta de los nombres y los significados de los términos. Establecen una terminología compartida entre miembros de una comunidad de interés. Estos miembros pueden ser humanos o *agentes* automáticos. A través de compromisos ontológicos compartidos, los agentes inteligentes pueden alcanzar un *conocimiento compartido parcial* y de esta forma interoperar.

La representación basada en marcos (frames) puede verse como un paradigma adecuado para la construcción de ontologías, y el formalismo RDFS del W3C (y sus extensiones, como DAML+OIL y OWL) puede verse como una manifestación de la representación basada en marcos para el Web. Los marcos están estrechamente relacionados a una técnica de KR anterior, las redes semánticas, que a su vez están basadas en la idea de memoria asociativa humana. Pueden entenderse sencillamente como estructuras del datos de nodos – “conceptos” – y enlaces – “asociaciones” – entre ellos.

²⁰ W3C. *Feature Synopsis for OWL Lite and OWL W3C Working Draft 29 July 2002*. Deborah L. McGuinness, Frank van Harmelen, (eds.). W3C, 29 de julio de 2002. Disponible en : <http://www.w3.org/TR/2002/WD-owl-features-20020729/>

²¹ Schreiber, Guus. *A UML Presentation Syntax for OWL Lite*. [Incomplete draft, intended for internal Webont discussions]. 3 de abril de 2002. Disponible en: <http://www.swi.psy.uva.nl/usr/Schreiber/docs/owl-uml/owl-uml.html>

²² W3C, RDFCore Working Group, <http://www.w3.org/2001/sw/RDFCore/>

²³ Lassila, O.; McGuinness, D. *The Role of Frame-Based Representation on the Semantic Web*. KSL Tech Report Number KSL-01-02, enero de 2001. Disponible en: <http://www.ksl.stanford.edu/people/dlm/etai/lassila-mcguinness-fbr-sw.html>

²⁴ Gruber, T. R. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220, 1993. Disponible en : http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html

La simplicidad y facilidad de entendimiento de los sistemas basados en marcos pueden convertirlos en el paradigma para la representación del conocimiento en entornos Web. Un ejemplo de un sistema basado en marcos es Ontolingua²⁵. Proporciona una sintaxis basada en marcos pero traduce toda la información en KIF que es simplemente una forma de codificación de la información en lógica de primer orden.

Los sistemas basados en marcos, con su íntimo parentesco con las ampliamente conocidas tecnologías orientadas a objetos, pueden demostrar ser el paradigma “adecuado”. Existe una conexión entre los sistemas de marcos y la programación orientada a objetos (OOP). El vocabulario básico es diferente, pero los términos denotan aproximadamente lo mismo:

Sistemas OOP	Sistemas de frames	Lógicas descriptivas
Instancia	Instancia, frame, individuo	Instancia, individuo
atributo, variable de instancia	Slot	rol
Valor	Filler	Filler
Clase, tipo	frame, esquema	clase, concepto

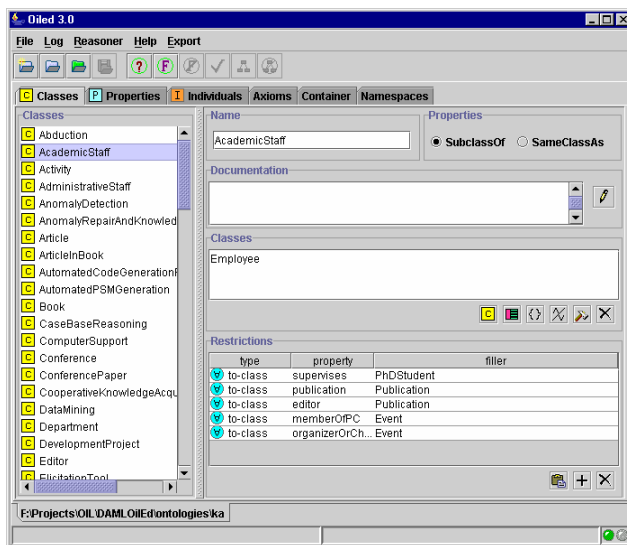


Fig. 4. Editor OilEd 3.0

Un ejemplo de editor basados en marcos es **OilEd**²⁶, un editor sencillo de ontologías. Se desarrolló inicialmente como una demostración de las posibilidades y beneficios de su uso como razonador para la clasificación de ontologías, pero ha alcanzado bastante éxito como editor de ontologías por la adopción de un paradigma "basado en marcos", estre-

²⁵ Farquhar, A.; Fikes, R.; Rice, J. The Ontolingua Server: a Tool for Collaborative Ontology Construction. *Intl. Journal of Human-Computer Studies*, 46, 1997.

²⁶ Bechhofer, S.; Horrocks, I.; Goble, C.; Stevens, R. OilEd: a Reason-able Ontology Editor for the Semantic Web. *Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence, September 19-21, Vienna*. Springer-Verlag LNAI v. 2174, pp. 396-408, 2001. Disponible en: <http://potato.cs.man.ac.uk/papers/ki2001.pdf>

chamente ligado a la descripción del lenguaje OIL subyacente. Este editor soporta algunos tipos de datos XMLS y puede exportar los esquemas en DAML+OIL, RDFS y HTML.

Otras herramientas como Protegé 2000²⁷ (con gran influencia en OilEd) y OntoEdit²⁸ también usan el paradigma de marcos.

En términos prácticos, el desarrollo de una ontología implica²⁹:

- Definir *clases* en la ontología
- Colocar las clases en un jerarquía de taxonomías (subclase-superclase)
- Definir slots [atributos] y describir los valores permitidos para esos slots
- Rellenar los valores de los slots con ejemplos.

UML (Unified Model Language)

UML se diseñó como lenguaje de modelado para el desarrollo de sistemas software y es ampliamente conocido y utilizado en ingeniería de software y modelado de datos. UML es un modelo conceptual propuesto como lenguaje de modelado de ontologías³⁰.

En resumen, UML tiene las siguientes características relevantes³¹:

- *Notación abstracta*: dispone de una notación gráfica de modelos legible por humanos. Un modelo UML es comparable a un esquema u ontología. Una instancia de un modelo UML comprende objetos que participan en varias relaciones con otros objetos.
- *Serialización XMI*: XML Metadata Interchange (XMI) es una codificación basada en XML para modelos UML.
- *Object Constraint Language (OCL)* es un lenguaje formal usado para especificar reglas de buena formación de modelos. Restringe el número de posibles instancias válidas de modelos UML.
- *UML CORBAfacility* es una API neutral para la manipulación de modelos UML.
- *Estructura de cuatro niveles*: la arquitectura UML se basa en una arquitectura de cuatro niveles. Los cuatro niveles son: objetos de usuario, modelo, metamodelo y meta-metamodelo. UML está interesado principalmente con el nivel de metamodelo, que es una instancia del nivel de meta-metamodelo. La capa meta-metamodelo se define en un estándar separado llamado MOF (Meta-Object Facility).

UML y RDFS

La aplicación de UML a cualquier tipo de modelo de datos requiere un “perfil de extensión”. Por ejemplo, hay múltiples perfiles y correspondencias entre UML y XML, pero no todas persiguen los mismos objetivos. Las especificaciones XML Metadata Interchange (XMI) y XMI Production for W3C XML Schema, de OMG, ofrecen un mapa de co-

²⁷ Grosso W. E. [et al.]. Knowledge modeling at the millennium (the design and evolution of Protegé 2000). *Proceedings of KAW99*, 1999.

²⁸ Staab, S.; Maedche, A. Ontology engineering beyond the modeling of concepts and relations. *Proceeding del ECAI'2000 Workshop on Application of Ontologies and Problem-Solving Methods*, 2000.

²⁹ Mulholland, P. *Introduction to Ontologies*. Knowledge Media Institute, 1999.

³⁰ Cranefield, S.; Purvis, M. UML as an ontology modelling language. *Proceedings of the Workshop on Intelligent Information Integration, 16^a International Joint Conference on Artificial Intelligence*, 1999.

³¹ Características comparables a la identidad de objeto y a las relaciones binarias de RDF se definen en el paquete UML Foundation/Core.

rrespondencias estándar desde UML a XMLS con el propósito de intercambiar modelos entre herramientas UML.

Provost analiza los problemas en la determinación de la correspondencia entre estos dos metamodelos y adelanta un “perfil de extensión” de UML que soporte la expresión completa de un conjunto de información en XMLS (vocabulario XML)³². La principal diferencia con XMI es que XMI pone a UML en primer lugar, y en algunos casos se establecen correspondencias que no capturan construcciones útiles de XMLS, aún cuando el modelo UML esté bien expresado. El objetivo es poner XMLS en primer lugar y desarrollar un perfil de extensión UML para el uso específico del diseño XMLS: el perfil de extensión debe capturar todos los detalles de un vocabulario XML que XMLS pueda expresar y debe soportar la generación bidireccional de documentos XMLS.

Una aproximación similar puede tomarse para el establecimiento de correspondencias entre UML y RDFS. Podría desarrollarse un perfil de extensión de UML que soporte la expresión completa del conjunto de información RDFS.

Como Chang expone en una nota publicada por el W3C, el modelo **RDFS es equivalente a un subconjunto del modelo de clases en UML**³³. RDFS usa un DLG (Directed Labeled Graph, Grafo dirigido etiquetado) para describir los esquemas. Los esquemas de clases expresados en UML también pueden verse como DLGs. Si se usa un DLG para un esquema del modelo de clases de UML, puede demostrarse que el DLG de RDFS es isomórfico a un subgrafo del DLG del esquema de clases de UML. Las estructuras y elementos en el modelo de clases de UML y de RDFS se corresponden unos con otros. El resto de la nota describe la relación entre los elementos de los modelos de clase para RDFS y UML. Un esquema RDFS puede transformarse a una representación equivalente en un esquema de clases UML. Dada una representación del esquema de clases de UML podemos especificar una representación equivalente en RDFS.

Editores basados en perfiles de extensión UML

Como ejemplo de herramienta de visualización y edición de ontologías basados en perfiles de extensión podemos considerar a DUET³⁴, el cual proporciona un entorno para la visualización, desarrollo y manipulación de ontologías DAML+OIL en Rational Rose 2000 y ArgoUML V0.10. En la documentación de DUET podemos encontrar las correspondencias específicas usadas por DUET para la transformación bidireccional entre DAML+OIL y UML.

³² Provost, W. *UML For W3C XML Schema Design*. XML.com, 7 de agosto de 2002. Disponible en: http://www.xml.com/pub/a/2002/08/07/wxs_uml.html

³³ Chang, W. *A Discussion of the Relationship Between RDF-Schema and UML*. W3C Note. W3C, 4 de agosto de 1998. Disponible en: <http://www.w3.org/TR/1998/NOTE-rdf-uml-19980804/>

³⁴ DUET v.0.3 (DAML UML Enhanced Tool), Programa CODIP (Components for Ontology Driven Information Push), <http://codip.grci.com/Tools/Tools.html>

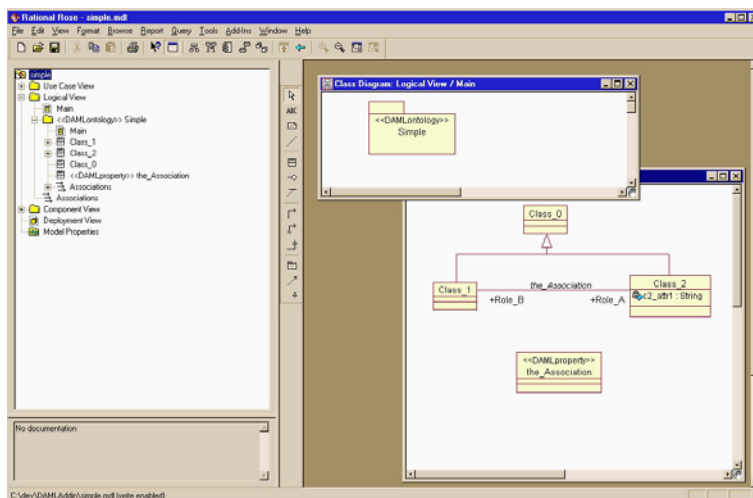


Fig. 5. DUET como perfil de extensión en Rational Rose

Interoperabilidad entre modelos e implementación

En el Web Semántico, la audiencia destino son las máquinas. Para satisfacer las demandas de esta audiencia, la información necesita estar disponible en formato procesable por máquina. Para hacer realidad este objetivo están disponibles una variedad de modelos de información como RDFS o UML, aunque varían en sus capacidades. XML propicia un acuerdo general en la sintaxis de la codificación de la información procesable por máquina. Sin embargo, conseguir interoperabilidad sintáctica entre diferentes modelos de información es una tarea laboriosa.

Melnik y Decker, proponen una aproximación en capas para facilitar la interoperabilidad de los modelos de información, tomada prestada de las técnicas de estructuración del software usadas en entornos de redes³⁵. Identifica lo que denomina la capa de objetos que llena el hueco entre la sintaxis y las capas semánticas. Sugiere que las características clave (primitivas de modelado) de la capa de objetos son: identificación de objetos y relaciones binarias, tipado básico, reificación, relaciones ordenadas y relaciones n-arias.

Finalmente examina aspectos de diseño e implementaciones alternativas (de las primitivas de modelado) relacionadas con la construcción de la capa de objetos. Por ejemplo, analiza las posibles implementaciones lógicas de las relaciones ordenadas (especialización, contenedor,...). De todas ellas, considera que la implementación contenedor es la menos significativa semánticamente.

³⁵ Melnik, S.; Decker, S.A. *Layered Approach to Information Modeling and Interoperability on the Web*. Stanford University, 4 de septiembre de 2000. Disponible en: <http://www-db.stanford.edu/~melnik/pub/sw00/sw00.pdf>

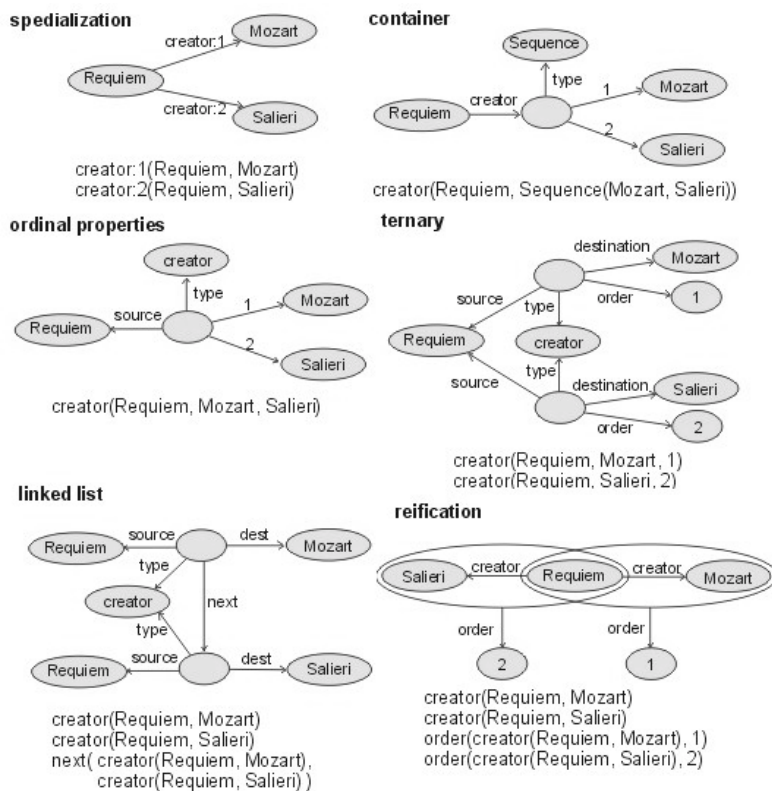


Fig. 6. Diferentes implementaciones lógicas de relaciones ordenadas

3.3. Perfiles de Aplicación (application profiles)

Un perfil de aplicación es un ensamblaje de elementos de metadatos seleccionados de uno o más esquemas de metadatos y combinados en un esquema compuesto. Los perfiles de aplicación permiten a los diseñadores la “mezcla y combinación” (“mix and match”) más apropiada de diferentes esquemas. Por definición, un perfil de aplicación no puede introducir nuevos elementos: cada elemento tiene que venir de un namespace particular y cuando se necesita un nuevo término, se tiene que crear un nuevo namespace para él³⁶.

El objetivo principal de los perfiles de la aplicación es aumentar la interoperabilidad semántica de los metadatos resultantes dentro de una comunidad, yendo más allá del acuerdo general universal de un solo estándar, sin comprometer la interoperabilidad básica que el estándar habilita en los límites de estas comunidades³⁷.

Para el diseño de un perfil de aplicación se requieren varios mecanismos:

³⁶ Heery, Rachel; Patel, Manjula. *Application Profiles: Mixing and Matching Metadata Schemas*. Ariadne, 25, Septiembre de 2000. Disponible en: <http://www.ariadne.ac.uk/issue25/app-profiles/intro.html>

³⁷ Special section on semantic interoperability in global information systems, *ACM SIGMOD Record*, v. 28, n. 1, marzo de 1999. Disponible en: <http://www.acm.org/sigmod/record/issues/9903/>

- **Especificación de cardinalidades:** la cardinalidad se refiere a las restricciones en la aparición de un elemento. Los elementos pueden declararse como **optativos, obligatorios o condicionales**. Cuando un perfil de aplicación debe operar dentro de las restricciones de interoperabilidad definidas por el estándar, no puede relajar el estado de los elementos de datos.
- **Restricción del espacio de valores:** Para algunos elementos, su espacio de valores puede hacerse más restrictivo que en el estándar. Aún cuando el estándar ya sea lo bastante explícito sobre el espacio de valores, el contexto de uso puede permitir nuevas restricciones. El uso de un **vocabulario controlado específico o namespace** precisará el alcance y aumentará la precisión de una descripción.
- **Especificación de relaciones y dependencias:** Un perfil de aplicación puede definir las relaciones mutuas entre los elementos y sus espacios de valores.

El etiquetado XML es todavía una parte pequeña del marcado global en el Web, pero es la opción preferida para la codificación e intercambio de datos estructurados. La especificación XML Esquema define un lenguaje de esquema que permite la especificación de *perfiles de aplicación* que permite la interoperabilidad entre aplicaciones.

Como hemos comentado anteriormente, las principales comunidades de metadatos están desarrollando sus perfiles de aplicación en XMLS. XMLS es estable, ampliamente aceptado en aplicaciones comerciales y proporciona los medios para implementar las propiedades descritas anteriormente. Revisando las especificaciones de perfiles, podemos encontrar fácilmente la justificación de la preferencia por XML frente a RDF: RDFS es el futuro, pero la realidad en este momento es XMLS. Un esquema RDF **debería** ser un conjunto de informaciones relativas a las clases de recursos que permita explicitar las relaciones que se establecen entre ellas, o bien **matizar el carácter obligatorio u opcional de las propiedades y otras restricciones como el número de ocurrencias**, etc.

El propósito primario de un perfil de aplicación RDFS es documental, permitiendo a los proveedores de información declarar sus modelos de metadatos de una manera uniforme, y que al mismo tiempo les permita su integración en un registro de esquemas y perfiles de aplicación. El tipo documental de perfil está en contraste con el tipo de perfil XMLS, cuyo propósito primario es validar la estructura de etiquetas de archivos de metadatos, y con cualquier otro tipo de perfil “operacional” directamente utilizable dentro de los entornos software para el procesamiento de metadatos.

De forma abstracta, algunas veces se usa “esquema” para designar un conjunto de unidades semánticas (e.d., elementos de metadatos o encabezamientos de la materia) junto con sus atributos, como el nombre, identificador, definición, o relación con otras unidades semánticas. En el Proyecto SCHEMAS, han evitado referirse genéricamente a estos conjuntos de conceptos como “esquemas” y prefieren el popular término de “vocabulario”. Más restringidamente, “esquema” puede referirse a la representación de un vocabulario en un formato particular, como un esquema RDF o un esquema relacional (un esquema semántico). Más específicamente, “esquema” puede referirse a un archivo que describe la estructura de etiquetas de un documento XML, como una DTD en XML (un esquema de documento)³⁸.

³⁸ Proyecto SCHEMAS. *The SCHEMAS Forum – a Retrospective Glossary*. UK Office for Library and Information Networking (UKOLN), Universidad de Bath, actualización de 6 de marzo de 2002. Disponible en: <http://www.schemas-forum.org/info-services/d74.htm>

3.4. Registros de metadatos

Los registros de metadatos representan un tema importante de investigación sobre la biblioteca digital en este momento. Conforme aumente el número de metadatos y esquemas de perfiles de aplicación diseñados para satisfacer las necesidades de comunidades particulares, análogamente aumentará la importancia de la gestión y descubrimiento de registros.

- Se espera que los registros proporcionen los medios para identificar y hacer referencia a esquemas y perfiles de aplicación, potencialmente incluyendo los medios para la establecer la correspondencia entre los diferentes esquemas.
- Además, se espera que estos registros contengan, o se enlacen a, importantes vocabularios controlados de los que puedan seleccionarse los valores de campos de los metadatos.

Se están desarrollando varios proyectos relacionados con el registro de metadatos. Dos de los más importantes son:

El proyecto SCHEMAS

Está siendo desarrollado actualmente por UKOLN (United Kingdom Office for Library and Information Networking) con financiación del Programa IST (Information Society Technologies)³⁹. Su objetivo es el desarrollo de una base de datos completa de esquemas RDF, perfiles de aplicación, y semántica relacionada. Ha sido utilizada por otros proyectos del Programa IST y otras iniciativas europeas relacionadas. La base de datos de SCHEMAS se usará para promover la reusabilidad e interoperabilidad semántica de proyectos nuevos y existentes.

El proyecto Open Metadata Registry

Este proyecto, desarrollado por la DCMI, tiene mucho en común con el proyecto SCHEMAS. Su objetivo, como el del proyecto SCHEMAS, es la creación de una base de datos de vocabularios y semántica relacionada pertenecientes a varias comunidades de recursos. También se usará para promover el descubrimiento y reusabilidad semántica de vocabularios existentes y la creación de nuevos vocabularios. La diferencia entre los dos, sin embargo, es que el proyecto Open Metadata Registry registrará vocabularios relacionados con la DCMI, mientras que el proyecto SCHEMAS registrará los esquemas RDF y namespaces usados por proyectos dentro de la Unión europea.

EOR puede considerarse un buen ejemplo de herramienta de fuente abierta para el desarrollo de aplicaciones RDF⁴⁰. Tanto SCHEMAS como DC Registry han construido prototipos basados en EOR.

Estos registros asumirán las características de un diccionario electrónico, disponible para la consulta de:

- **Diseñadores de aplicaciones** que podrán consultar los registros para identificar los esquemas de metadatos existentes y componentes del esquema que podrían satisfacer sus necesidades o identificar las extensiones a aquéllos esquema que otros diseñadores han desarrollado para satisfacer una determinada necesidad local.

³⁹ SCHEMAS Project, <http://www.schemas-forum.org/>

⁴⁰ The EOR Toolkit, <http://eor.dublincore.org/>

- Los **creadores y gestores de metadatos** que pueden consultar un registro para determinar la definición o especificaciones de uso acerca de un elemento o los conjuntos de valores candidatos disponibles o preferidos a utilizar para poblar los elementos particulares.
- **Aplicaciones** que pueden resolver URIs asociadas con un esquema, un elemento, o un conjunto de valores para comparar o evaluar elementos o sus valores en un conjunto de metadatos.
- **Usuarios finales** que podrían consultar un registro para entender bien las definiciones o contexto de términos de metadatos y que podrían mejorar la efectividad de su búsqueda o procesamiento.

Así, los registros proporcionarán los medios para gestionar y descubrir declaraciones de esquema de metadatos, declaraciones de perfil de aplicación, y declaraciones de espacios de valores. Como cualquier esquema de metadatos o perfil de la aplicación evoluciona, los registros mantendrán las relaciones entre las diversas versiones de ese esquema para promover la interoperabilidad semántica y de máquina en el tiempo⁴¹.

El Grupo de trabajo *DCMI Registry* está explorando algunos de estos problemas a través de la explicación de requisitos funcionales para un registro de metadatos DCMI plurilingüe y un sistema de gestión de vocabulario. Los prototipos iniciales para este sistema pueden accederse en el web de DCMI⁴².

Probablemente los registros variarán en su funcionalidad con algunos con enlaces simples a las declaraciones del esquema mientras otros pueden ser ricas bases de datos funcionales. Algunos registros se gestionarán por las autoridades del namespace y mantendrán copias canónicas del esquema y declaraciones de espacios de valores mientras que otros registros recopilarán declaraciones de esas fuentes autoritativas y los harán disponibles de una manera distribuida.

4. CONCLUSIONES

Los metadatos son una parte clave de la infraestructura de información necesaria para ayudar a crear orden en el caos del Web, infundiendo descripción, clasificación, y organización que ayuden a la creación de almacenes de información más útiles. Los orígenes de metadatos, como los orígenes de los recursos mismos, serán de calidad diferente y se organizarán alrededor de propósitos diferentes para reflejar los objetivos de los proveedores de información. Para que estas oportunidades lleguen a ser una realidad, será necesaria cierta convergencia de formatos de codificación y una semántica comúnmente acordada.

RDF *promete* una arquitectura de metadatos Web y ha estado desarrollándose como la principal “infraestructura habilitadora” de la actividad sobre Web Semántica en el W3C. Diseñado para promover la reutilización e intercambio de vocabularios, RDF es una capa adicional (semántica) sobre XML (sintáctica) que simplifica la reutilización de términos de vocabulario entre namespaces. La despliegue de RDF ha sido hasta ahora *experimental*, aunque están surgiendo importantes aplicaciones.

RDF proporciona soluciones que habilitarán un mayor grado de fiabilidad, relevancia, y exactitud de las aplicaciones y servicios orientados al descubrimiento del recurso y a la

⁴¹ Heery, Rachel; Patel, Manjula. Application Profiles: Mixing and Matching Metadata Schemas. *Ariadne*, 25, Septiembre de 2000. Disponible en: <http://www.ariadne.ac.uk/issue25/app-profiles/intro.html>

⁴² The Open Metadata Registry Prototype [Home Page]. <<http://wip.dublincore.org:8080/registry/Registry>>

gestión de sitios Web y otros recursos de Internet: Semántica entendible por máquina que posibilitará el descubrimiento automatizado, gestión, e intercambio de recursos; un mayor grado de detalle en la descripción (granularidad, nivel de detalle al que se ve o describe un objeto de información o recurso) y de precisión para el descubrimiento de recursos; y la interoperabilidad semántica entre diferentes vocabularios de metadatos que usan una sintaxis común.

Se considera a RDFS (también a UML) como un modelo conceptual apropiado para la descripción de recursos. Sin embargo, se considera un error el énfasis en su sintaxis RDF/XML. RDFS como modelo de datos es más importante que su sintaxis y no debe ser confundido con ella.

Es necesario continuar con los esfuerzos de determinación de las primitivas básicas de modelado proporcionadas por RDFS, OWL, UML y los sistemas de marcos, así como determinar la mejor forma de implementación de las mismas como base para el establecimiento de mapas de correspondencias entre los distintos modelos conceptuales del Web semántico. Esta determinación permitirá rellenar el “hueco” entre el nivel sintáctico y semántico que está limitando el desarrollo e implementación de nuevas herramientas y aplicaciones que soporten RDF y RDF/XML.

UML es un estándar de modelado conceptual. Será necesario desarrollar “perfiles de extensión” que representen de la forma más conveniente las construcciones de RDFS y OWL. A diferencia de RDFS u OWL, es ampliamente conocido por cualquier estudiante de Informática y dispone de herramientas CASE de diseño como Rational Rose,...

Las principales comunidades de metadatos están desarrollando sus perfiles de aplicación en XMLS. XMLS es estable, ampliamente aceptado en aplicaciones comerciales y proporciona los medios para implementar perfiles de aplicación y aplicaciones basadas en ellos. Revisando las especificaciones de perfiles, podemos encontrar fácilmente la justificación de la preferencia por XMLS: RDFS/OWL es el futuro, pero en este momento la realidad es XMLS.

Se están desarrollando esfuerzos interesantes en la determinación y entendimiento de los elementos comunes en los distintos esquemas, y en el diseño de ontologías y metamodelos que acomoden la noción de esquema base para la interoperabilidad semántica entre diferentes dominios. Los registros de esquemas y perfiles de aplicación constituyen una herramienta valiosa para la determinación de mapas de correspondencias entre los diferentes esquemas de metadatos.

5. REFERENCIAS

- Bechhofer, S.; Horrocks, I.; Goble, C.; Stevens, R. OilEd: a Reason-able Ontology Editor for the Semantic Web. *Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence, September 19-21, Vienna*. Springer-Verlag LNAI v. 2174, pp. 396-408, 2001. Disponible en: <http://potato.cs.man.ac.uk/papers/ki2001.pdf>
- Chang, W. *A Discussion of the Relationship Between RDF-Schema and UML*. W3C Note. W3C, 4 de agosto de 1998. Disponible en: <http://www.w3.org/TR/1998/NOTE-rdf-uml-19980804/>

- Cranefield, S.; Purvis, M. UML as an ontology modelling language. Proceedings of the Workshop on Intelligent Information Integration, *16^a International Joint Conference on Artificial Intelligence*, 1999.
- Cross, P.; Brickley, D. Conceptual relationships for encoding thesauri, classification systems and organised metadata collections and a proposal for encoding a core set of thesaurus relationships using an RDF Schema. *Institute for Learning & Research Technology, University of Bristol, UK.*, 6 de junio de 2002. Disponible en: <http://www.desire.org/results/discovery/rdftheschema.html>
- Duval, E.; Hodgins, W.; Sutton, S.; Weibel, S. Metadata Principles and Practicalities. *D-Lib Magazine*, v.8, n.4, abril de 2002. Disponible en: <http://www.dlib.org/dlib/april02/weibel/04weibel.html>
- Farquhar, A.; Fikes, R.; Rice, J. The Ontolingua Server: a Tool for Collaborative Ontology Construction. *Intl. Journal of Human-Computer Studies*, 46, 1997.
- Grosso W. E. [et al.]. Knowledge modeling at the millennium (the design and evolution of Protegé 2000). *Proceedings of KAW99*, 1999.
- Gruber, T. R. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220, 1993. Disponible en: http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html
- DCMI. *Guidance on expressing the Dublin Core within the Resource Description Framework (RDF). Draft Proposal*. Eric Millar, Paul Millar y Dan Brickley, eds. DCMI, 1 de julio de 1999. Disponible en <http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/>
- DCMI. *Guidelines for implementing Dublin Core in XML*. DCMI, Proposed Recommendation. A. Powell y P. Johnston, eds. DCMI, 23 de Julio de 2002. Disponible en: <http://www.ukoln.ac.uk/documents/2002/07/23/dc-xml-guidelines/>
- Heery, Rachel; Patel, Manjula. Application Profiles: Mixing and Matching Metadata Schemas. *Ariadne*, 25, Septiembre de 2000. Disponible en: <http://www.ariadne.ac.uk/issue25/app-profiles/intro.html>
- Lassila, O.; McGuinness, D. The Role of Frame-Based Representation on the Semantic Web. *KSL Tech Report Number KSL-01-02*, enero de 2001. Disponible en: <http://www.ksl.stanford.edu/people/dlm/etai/lassila-mcguinness-fbr-sw.html>
- Melnik, S.; Decker, S.A. Layered Approach to Information Modeling and Interoperability on the Web. *Stanford University*, 4 de septiembre de 2000. Disponible en: <http://www-db.stanford.edu/~melnik/pub/sw00/sw00.pdf>
- Mulholland, P. Introduction to Ontologies. *Knowledge Media Institute*, 1999.
- Patel-Schneider, P.; Siméon, J. The Ying/Yang Web: XML syntax and RDF semantics. *WWW2002*, Honolulu, Hawai, mayo de 2002. Disponible en: <http://www2002.org/CDROM/refereed/231/index.html>
- Provost, W. UML For W3C XML Schema Design. *XML.com*, 7 de agosto de 2002. Disponible en: http://www.xml.com/pub/a/2002/08/07/wxs_uml.html
- Proyecto SCHEMAS. The SCHEMAS Forum – a Retrospective Glossary. *UK Office for Library and Information Networking (UKOLN)*, Universidad de Bath, actualización de 6 de marzo de 2002. Disponible en: <http://www.schemas-forum.org/info-services/d74.htm>

- Schreiber, Guus. *A UML Presentation Syntax for OWL Lite*. [Incomplete draft, intended for internal Webont discussions]. 3 de abril de 2002. Disponible en: <http://www.swi.psy.uva.nl/usr/Schreiber/docs/owl-uml/owl-uml.html>
- Special section on semantic interoperability in global information systems, *ACM SIGMOD Record*, v. 28, n. 1, marzo de 1999. Disponible en: <http://www.acm.org/sigmod/record/issues/9903/>
- Staab, S.; Maedche, A. Ontology engineering beyond the modeling of concepts and relations. *Proceeding del ECAI'2000 Workshop on Application of Ontologies and Problem-Solving Methods*, 2000.
- W3C. *Namespaces in XML*. Tim Bray, Dave Hollander, y Andrew Layman (eds.). 14 de enero de 1999. Disponible en: <http://www.w3.org/TR/REC-xml-names>
- W3C. *RDF/XML Syntax Specification (Revised)*. W3C Working Draft 25 March 2002. Dave Beckett, ed. W3C, 25 de marzo de 2002. Disponible en: <http://www.w3.org/TR/2002/WD-rdf-syntax-grammar-20020325>
- W3C. *Resource Description Framework (RDF): Model and Syntax Specification*. W3C Recommendation, 22 February 1999. [Ora Lassila y Ralph R. Swich, eds. Disponible en: <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>
- W3C. *Feature Synopsis for OWL Lite and OWL* W3C Working Draft 29 July 2002. Deborah L. McGuinness, Frank van Harmelen, (eds.). W3C, 29 de julio de 2002. Disponible en: <http://www.w3.org/TR/2002/WD-owl-features-20020729/>
- W3C. *OWL Web Ontology Language 1.0 Reference*. W3C Working Draft 29 July 2002. Mike Dean, Dan Connolly, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider y Lynn Andrea Stein, eds. W3C, 29 de julio de 2002. Disponible en: <http://www.w3.org/TR/2002/WD-owl-ref-20020729/>

Noviembre de 2002