FLERSA: Un Sistema Semántico de Gestión de Contenido Web (S-CMS)

TESIS DOCTORAL

José Luis Navarro Galindo

Programa de Doctorado sobre Métodos y Técnicas Avanzadas del Desarrollo de Software (125.99.1)

Departamento de Lenguajes y Sistemas Informáticos
Escuela Técnica Superior de Informática y Telecomunicaciones
Universidad de Granada

Mayo 2012

Editor: Editorial de la Universidad de Granada
Autor: José Luis Navarro Galindo
D.L.: GR 211-2013
ISBN: 978-84-9028-281-6

FLERSA: Un Sistema Semántico de Gestión de Contenido Web (S-CMS)

Memoria que presenta para optar al Título de Doctor en Informática

José Luis Navarro Galindo

Dirigida por el Doctor

José Samos Jiménez

Programa de Doctorado sobre Métodos y Técnicas Avanzadas del Desarrollo de Software (125.99.1)

Departamento de Lenguajes y Sistemas Informáticos

Escuela Técnica Superior de Informática y Telecomunicaciones

Universidad de Granada

Mayo 2012

ISBN XXX-XX-XXX-XXXX-X

El doctorando José Luis Navarro Galindo y el director de la tesis José Samos Jiménez. Garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección del director de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada,

Director de la Tesis	Doctorando
Fdo.:	Fdo.:

Dedicada a mis padres José y María Luisa,
y a mi hermana Rosa, por el apoyo
y confianza que depositan en mí siempre
que emprendo un proyecto en la vida.

Agradecimientos

No nos falta valor para emprender
ciertas cosas porque son difíciles, sino
que son difíciles porque nos falta valor
para emprenderlas.

Séneca.

Cuando comencé mis estudios de Ingeniería Informática, allá por el año 1994, no tenía muy claro si sería capaz de terminar la carrera, debido a la dificultad que encontraba en el nuevo y vasto mundo que suponían las ciencias de la computación para mí. Al parecer no se me ha dado tan mal como pensaba inicialmente y aquí me encuentro, escribiendo las últimas líneas de mi tesis doctoral ... quién me lo iba a decir.

Desde aquí, quisiera agradecer por sus enseñanzas, a todas las personas que han dejado huella en mí a lo largo de las diferentes etapas de mi formación, porque soy consciente de que he llegado hasta aquí gracias a ellos. En particular quiero expresar mi agradecimiento a los maestros del C.P. "*San Andrés*" de Montejícar por iniciar mis inquietudes intelectuales; a los padres Agustinos del Colegio "*Nuestra Señora del Buen Consejo*" por inculcarme un saludable hábito de estudio; al profesorado del I.E.S. "*Francisco Ayala*" por la inmejorable preparación pre-universitaria que me dieron; al profesorado de la Escuela Técnica de Ingeniería Informática de Granada por los conocimientos que me han transmitido; a mis compañeros Hewlett-Packard y de los Servicios de Informática del Hospital Virgen de las Nieves por enseñarme importantes facetas del mundo laboral; a mis compañeros de secundaria de los distintos centros por los que he pasado (Fuentezuelas, Ilíberis y Zaidín-Vergeles) por los buenos ratos que pasamos y por todo lo que estoy aprendiendo de ellos en el mundo de la enseñanza.

Por último, expresar también mi agradecimiento a José Samos, mi director de tesis, por apostar por mí en este proyecto, por sus orientaciones cuando "*los árboles no me dejan ver el bosque*" y por la ayuda prestada durante todo este tiempo.

Resumen

La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica.

Aristóteles

En este trabajo, se presenta FLERSA (FLExible Range Semantic Annotation) como una herramienta de anotación semántica de contenido Web centrada en el usuario. La herramienta ha sido desarrollada a partir de un WCMS (Web Content Management System) y su principal objetivo es convertir la infraestructura específica de los WCMS en su equivalente semántico, extendiendo así los beneficios de la Web Semántica. Los principios y técnicas de FLERSA pueden aplicarse a cualquier WCMS.

La herramienta permite anotaciones semánticas manuales y automáticas, así como funciones de búsqueda mejoradas. Las anotaciones se basan en la ontología FLERSA-ontology, se trata de una "*ontología base*" inspirada en el marco de trabajo Annotea, cuyo propósito principal es dar soporte para la definición de anotaciones que se usan a modo de infraestructura; se pueden definir anotaciones adicionales a partir de ellas, usando conceptos y propiedades de otras ontologías.

Para la anotación semántica manual, se ha usado una nueva técnica de marcado de rangos flexibles, basada en el estándar RDFa, con la ventaja de que soporta la evolución de los documentos web que se anotan semánticamente más efectivamente que otras técnicas como pueden ser XPointer.

Para la anotación semántica automática, se ha usado un enfoque híbrido basado en técnicas de aprendizaje automático tales como el Modelo de Espacio Vectorial y N-gramas, para determinar los conceptos que se tratan en el contenido de un documento web. Los conceptos se organizan en torno a una taxonomía proporcionada por una ontología. La técnica de aprendizaje automático se basa en anotaciones previas que se usan a modo de Corpus.

En cuanto a las funciones de búsqueda mejoradas, comentar que el objetivo de la herramienta es explotar la información semántica de las anotaciones para conseguir resultados "*inteligentes*" en respuesta a las consultas.

Se realiza un doble almacenamiento de las anotaciones: en el servidor, en formato RDF, e incrustadas dentro del documento web donde se realizan, en formato RDFa, de forma totalmente transparente a los usuarios. Esta característica combina las ventajas del almacenamiento centralizado de anotaciones con aquellas del modelo incrustado como son:

- Permite inferir nuevo conocimiento a partir de la base de datos de anotaciones.
- Disponibilidad de las anotaciones semánticas autocontenidas dentro del documento.
- Acceso libre a los metadatos para motores de búsqueda web y otros tipos de servicios web, con objeto de mejorar las búsquedas.
- Proporciona información de la estructura interna de los documentos y de relaciones entre ellos.

Cabe destacar, que tanto el uso de ontologías como elemento de representación de conocimiento consensuado, así como las tecnologías emergentes recomendadas por la W3C, tales como XML, RDF, RDFa y OWL, han estado siempre presentes y han jugado un papel central durante todo el proceso de diseño e implementación de la herramienta FLERSA.

El resultado de todo el trabajo realizado se materializa en una extensión de Joomla. Se ha desarrollado un componente llamado `com_semantic` que está disponible libremente desde la URL `http://salmer.sourceforge.net`. Se ha publicado bajo licencia Affero GNU/GPL v3 y proporciona la implementación del sistema de anotación semántica que se describe en el presente trabajo.

Índice

Índice de figuras

Índice de Tablas

Capítulo 1

Introducción

Quédate ante la puerta si quieres que te la abran. No dejes el camino si quieres que te guíen. Nada está nunca cerrado sino a tus propios ojos.

Attar Farid

1.1. Introducción

En los últimos años se ha desarrollado el concepto de CMS (*Content Management System*, Sistema de Gestión de Contenido). Un CMS es una herramienta que permite crear y mantener un sitio web con facilidad, sin requerir conocimientos técnicos avanzados. En la actualidad existe una gran comunidad de usuarios que utilizan CMS de código abierto para el desarrollo de sitios web personales, de empresas e instituciones.

Por otro lado, las tecnologías emergentes de la Web Semántica (XML, RDF y OWL) aportan descripciones explícitas de los recursos de la Web (entre otros: Artículos, catálogos, formularios, y mapas) con objeto de facilitar la interpretación de los documentos y de realizar procesos inteligentes de captura y tratamiento de información (para permitir la interoperabilidad).

El proyecto que se presenta tiene como principal objetivo enriquecer el concepto de CMS con tecnologías de la Web Semántica y obtener así un S-CMS (*Semantic Content Management System*, Sistema Semántico de Gestión de Contenido). Aunque el término genérico S-CMS es usado por algunos autores, hasta ahora, las tecnologías y principios que se aplican bajo esa denominación generalmente no son los de la Web Semántica.

Varios CMS se autoproclaman como materialización del concepto de la

Web 2.0, por otra parte, se empieza a hablar de la Web 3.0 generalmente identificando el término con la Web Semántica, que algunos tachan como "inalcanzable". Así, podríamos definir un S-CMS como un "puente" entre la Web 2.0 y la Web 3.0.

1.2. Objetivos de la Tesis

El principal objetivo es definir y desarrollar el concepto de **S-CMS** para paliar en parte las carencias de contenido semántico de la Web actual y para aportar a los CMS los múltiples beneficios que ofrecen las tecnologías de la Web Semántica (por ejemplo proporcionar búsquedas semánticas de contenidos o facilitar la integración y la interoperabilidad con otros S-CMS). Para ello, habrá que adaptar las tecnologías de la Web Semántica a los sistemas actuales.

La propuesta de S-CMS pretende ser lo suficientemente general como para que cualquier CMS se pueda convertir a su equivalente semántico; asimismo, un objetivo concreto del proyecto es aplicarla como prueba de concepto a un CMS de código abierto de amplia difusión y ofrecer el resultado a la comunidad de usuarios también como código abierto.

Otro de los objetivos es que los S-CMS puedan ser gestionados por usuarios con conocimientos técnicos similares a los actuales gestores de CMS.

1.3. Estructura de la Tesis

A continuación se presentan las etapas que han tenido lugar a lo largo del trabajo que ha dado como resultado la presente tesis. En cada una de ellas se realiza una pequeña descripción de la dirección en la que se encontraba la tesis; también se especifica el capítulo o capítulos de la tesis donde se desarrollan en profundidad. Son las que se describen a continuación:

- **Definición del problema:** Como queda patente en la introducción anterior, la Web actual tiene carencias de contenido semántico. Nos proponemos enriquecer los CMS actuales mediante el uso de recursos tales como ontologías y tecnologías emergentes, para obtener así los S-CMS, sus equivalentes semánticos.

- **Estado del arte:** Se trata de una etapa inicial de investigación y estudio donde se trabaja principalmente en tres direcciones: La Web Semántica, los CMS en general y las anotaciones semánticas.

En cuanto a la Web Semántica, se realiza un estudio de las tecnologías existentes dentro de este ámbito haciendo especial hincapié en las dedicadas a trabajar con **ontologías**. Dicho estudio se puede encontrar en el capítulo 3 de la presente tesis. Fruto de este trabajo es el artículo titulado "Una Panorámica Actual de Software para trabajar con Ontologías" (Navarro-Galindo y Samos, 2007b).

Otra dirección de interés es el estudio de los CMS de código abierto que se están utilizando actualmente. En el capítulo 2 se realiza un análisis de sus tipos y de la funcionalidad que ofrecen, así como una evaluación de los mismos acorde con una serie de requisitos cuyo objetivo es la selección de un candidato donde realizar futuros desarrollos.

El capítulo 4 está dedicada al estudio de los aspectos relativos al marcado de texto y al proceso de anotación semántica del mismo, junto con las tecnologías asociadas.

- **Propuesta de solución:** Se trata del núcleo de la tesis y se desarrolla en el capítulo 5, el más extenso de los que la componen. Comienza estableciendo los objetivos que se pretenden alcanzar a la hora de definir un S-CMS. A continuación se especifican los requisitos de diseño del mismo y se establece la arquitectura semántica con la que se pretende enriquecer los CMS. También cuenta con subsecciones específicas dedicadas a estudiar el papel de las ontologías en esta arquitectura y la técnica de definición de rangos flexibles de texto. El capítulo concluye con el estudio detallado de como tienen lugar los procesos de anotación semántica tanto manual como automático.

  Derivado de todo este trabajo de investigación se han presentado los siguientes artículos:

  - "Flexible Range Semantic Annotations based on RDFa" (Navarro-Galindo y Samos, 2010a)
  - "Manual and Automatic Semantic Annotation of Web Documents: The FLERSA Tool" (Navarro-Galindo y Samos, 2010b)
  - "FLERSA: Soporte a la Definición de Anotaciones y Búsquedas Semánticas en un CMS" (Navarro-Galindo y Samos, 2011)
  - "The FLERSA Tool: Adding Semantics to a Web Content Management System" (Navarro-Galindo y Samos, 2012)

- **Prototipado:** Se trata de una etapa donde se realiza una prueba de concepto. La propuesta de solución de la etapa anterior se materializa en el desarrollo de una extensión para un conocido CMS. El CMS se llama Joomla y la extensión que se ha implementado recibe el nombre de `com_semantic`[1], acorde con las tecnologías actuales de programación

[1] `http://www.sourceforge.net/salmer`

Web y con los estándares propuestos por la W3C para la Web Semántica. El resultado es un prototipo[2] totalmente operativo que ilustra todos los aspectos que se han estudiado en la propuesta de solución. En el capítulo 6 se pueden encontrar todo tipo de detalles sobre su desarrollo, funcionalidades y descripción de su entorno de pruebas.

- **Evaluación de la solución:** A continuación, en el capítulo 7, se realiza un estudio comparativo de las funcionalidades que aporta la herramienta desarrollada con respecto a otras herramientas. También se lleva a cabo un estudio experimental del proceso de anotación semántica automático para determinar su efectividad.

  El trabajo de evaluación realizado a contribuido a la elaboración y presentación de los siguientes artículos:

  - "The FLERSA Tool: Adding Semantics to a Web Content Management System" (Navarro-Galindo y Samos, 2012)
  - "A Hybrid Approach to Text Categorization Applied to Semantic Annotation" (en revisión).

  Conclusiones y trabajo futuro: Por último, en el capítulo 8, se pueden encontrar el capítulo resumen de toda la tesis donde se explica el objeto de la misma y las aportaciones que se realizan en ella. También se exponen las distintas posibilidades de trabajo futuro relacionadas con el contenido de la tesis y con la línea investigadora que sigue.

1.4. Contribuciones

El desarrollo del trabajo que se presenta en esta tesis ha dado como fruto el nacimiento de un nuevo tipo de CMS: El Sistema Semántico de Gestión de Contenido Web o S-CMS. Cumple un doble cometido: Por un lado, adaptar las tecnologías emergentes de la Web Semántica para su uso por la comunidad de usuarios de CMS; y por otro, enriquecer los CMS con los beneficios que la Web Semántica puede aportar.

A continuación se describen, de forma más detallada, las diversas contribuciones que aporta la presente tesis al campo de la Web Semántica en el contexto de los CMS. Son las siguientes:

- Se ha definido una nueva forma de definición de anotaciones semánticas incrustadas dentro del mismo código HTML que se anota; está basada en uso de las tecnologías DOM nivel 2 (Kesselman et al., 2000), junto con los lenguajes de anotación RDF (Manola y Miller, 2004) y

[2] `http://www.scms.es`

RDFa (Adida y Birbeck, 2007). La técnica recibe el nombre de "Técnica de definición de rangos flexibles" (Navarro-Galindo y Samos, 2010a) porque permite la anotación semántica mediante la creación de rangos (fragmentos de código HTML) de tamaño variable e incluso en los que pueden aparecer incrustados elementos multimedia. Entre las diversas ventajas que proporciona esta técnica destacan:

  - Permite una evolución consistente de las anotaciones definidas por medio de esta técnica.
  - Evita el problema de la "Web Profunda" para las anotaciones semánticas definidas mediante esta técnica. Los indexadores de los motores de búsqueda Web tienen acceso a la información semántica almacenada en los documentos que usan esta técnica de anotación.
  - Permite el almacenamiento dual de las anotaciones semánticas definidas en un documento, tanto en el lado servidor como en el lado cliente, dentro del documento que se anota.

- Ilustra cómo usar ontologías con diferentes fines: Como infraestructura base para la anotación semántica y como taxonomía de conceptos que permite extender los vocabularios de dichas anotaciones (Navarro-Galindo y Samos, 2012).

- Desarrolla un enfoque híbrido para realizar la anotación semántica automática. El enfoque combina el Modelo de Espacio Vectorial con N-gramas para realizar de forma automática anotaciones semánticas donde se determinan los conceptos de los que trata un documento (Navarro-Galindo y Samos, 2010b).

- Aporta diversas modalidades de búsqueda local donde explotar la semántica de las anotaciones introducidas en los documentos con objeto de mejorar las búsquedas de información (Navarro-Galindo y Samos, 2011).

1.5. Conclusiones

La tesis desarrolla al concepto de **S-CMS**, el cual se propone como solución para paliar las carencias de contenido semántico de los CMS actuales y para aportarles los beneficios que ofrece la Web Semántica.

En este capítulo se ha visto cómo se estructura la tesis, estableciéndose una correspondencia entre las secciones de la misma y los trabajos que se han ido realizando en distintos ámbitos. En resumen son: Estudio del estado del arte, propuesta de solución, evaluación de la solución y conclusiones.

Destacar en particular que la solución propuesta se aborda a dos niveles: A nivel teórico en la presente tesis y en las publicaciones realizadas; y a nivel práctico mediante el desarrollo de un componente y la creación de un portal web dotado con un prototipo ilustrativo del concepto.

Para concluir, se han explicado de forma introductoria y resumida las diferentes contribuciones que ofrece la presente tesis.

Capítulo 2

Sistemas de Gestión de Contenido

No basta saber, se debe también aplicar.
No es suficiente querer, se debe también hacer.

Johann Wolfgang Goethe

Resumen: Este capítulo está dedicado a realizar una introducción a los Sistemas de Gestión de Contenido (CMS), y en concreto a los de contenido web (WCMS), ya que su uso está muy extendido y han tomado gran relevancia en nuestros días debido a que muchas empresas y particulares desarrollan sus portales web usándolos como infraestructura. En particular Joomla, uno de estos sistemas, es sobre el que se centra el presente trabajado.

2.1. Introducción

Los sitios web se han ido multiplicando exponencialmente durante el tiempo y, pese a que son muy útiles, están muy lejos de ser perfectos.

Durante mucho tiempo, empresas e instituciones han realizado grandes esfuerzos en el mantenimiento de sus sitios web. Los problemas más habituales con los que se enfrentaban han sido:

- Información obsoleta. Los contenidos son antiguos o inadecuados.
- Dificultad para localizar información.

- La actualización del sitio es compleja y en ocasiones se realiza tardíamente.
- Falta de control sobre el diseño y la navegación del sitio web.
- Falta de autoridad y de entendimiento con respecto al administrador web del sitio.

La Web de finales del siglo XX se caracterizaba por el uso exclusivo de soluciones manuales para su mantenimiento; el éxito y sostenibilidad de la Web del siglo XXI se ha conseguido gracias a su automatización mediante el uso del CMS.

2.2. Definición

CMS (Boiko, 2001) son las siglas de *Content Management System*, que se traducen en español como *Sistema de Gestión de Contenidos*. En general, un CMS es una aplicación informática que se usa para editar, administrar, publicar y realizar búsquedas sobre distintos tipos de medios digitales y textos electrónicos.

En el pasado, realizar un sitio web desde cero podía ser una labor complicada y laboriosa. Las herramientas para trabajar con sitios web estaban enfocadas a la creación y edición de contenido web y dejaban descuidadas las tareas de mantenimiento. Por esta razón, ha aparecido, en los últimos años, un término más específico: El WCMS (*Web Content Management System*, Sistema de Gestión de Contenido Web). Un WCMS es una aplicación web diseñada para facilitar la gestión de sitios web a usuarios sin conocimientos técnicos. Abarca el ciclo de vida completo de las páginas de un sitio web, desde herramientas simples de creación de contenidos, hasta la publicación y finalmente el archivo de contenidos obsoletos. Además de ayudar en la edición de contenidos, los WCMS también aportan herramientas de ayuda para tareas tales como: Generación de elementos de navegación del sitio, búsqueda e indexación de información, y gestión de permisos de usuario.

Se pueden obtener muchos beneficios mediante la puesta en marcha de un sitio web gestionado por un WCMS subyacente. Entre los más importantes cabe destacar:

- Proceso de autoría de contenidos simplificado.
- Mejora en los tiempos de creación y cambio en los contenidos.
- Mejor consistencia.

- Mejora en la navegación.
- Incremento en la flexibilidad
- Soporte para la creación de contenidos descentralizada.
- Incremento en la seguridad.
- Reducción en la duplicidad de información.
- Mayor capacidad de crecimiento.
- Reducción de los costes de mantenimiento.

Además de todos los anteriores, el mayor beneficio que un WCMS puede proporcionar a un sitio web es la ayuda en los objetivos y estrategias de negocio. Por ejemplo, el WCMS puede ayudar a mejorar las ventas, a aumentar la satisfacción de los clientes o ayudar en la comunicación con el público.

Destacar que esta tesis se centra en el uso más común del término CMS: El gestionar contenido web. En adelante se usará el término genérico CMS para referirse al tipo de sistemas WCMS; aunque semánticamente no es del todo correcto, el término CMS es el que se encuentra actualmente más difundido para referirse a este tipo de sistemas.

2.3. Funcionalidad

A nivel general, la funcionalidad que ofrece CMS se puede descomponer en las siguientes categorías principales:

- Creación de contenidos.
- Gestión de contenidos.
- Publicación.
- Presentación.

Las categorías se relacionan entre sí de la forma que se muestra en la figura 2.1. Cada una de éstas se va a estudiar más detenidamente en las siguientes subsecciones.

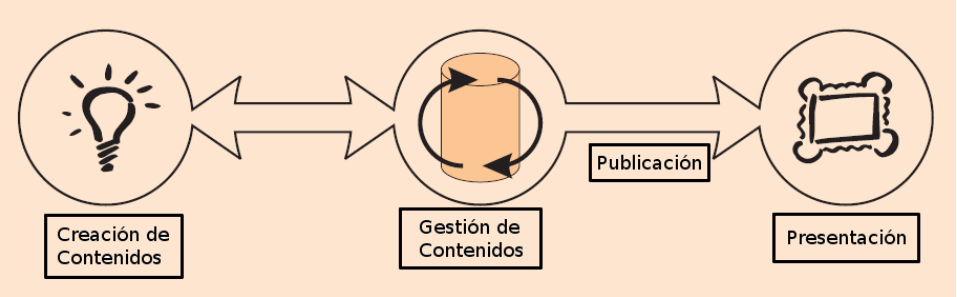


Figura 2.1: Funcionalidad de un CMS.

2.3.1. Creación de Contenidos

De cara al usuario, los sistemas CMS proporcionan herramientas de edición tipo WYSIWYG (*What You See Is What You Get*) mediante las cuales se puede crear o actualizar contenido web sin necesidad de poseer conocimientos técnicos (como por ejemplo conocer el lenguaje HTML). Casi todos los CMS actuales proporcionan un entorno de edición web, lo cual simplifica su implementación y permite la actualización remota de contenidos. Esta herramienta de edición es la clave del éxito que están teniendo los CMS: Proporciona un mecanismo sencillo para mantener el sitio web de forma que la edición y actualización puede transferirse a los usuarios. Por ejemplo, un comercial de una empresa puede mantener la sección de productos actualizada, mientras la dirección se encarga de la sección de prensa.

También suelen contar con herramientas para la definición de la estructura del sitio web; es decir, para determinar dónde va cada página y cómo se enlaza con las demás páginas. Algunas funcionan tan bien que ofrecen la posibilidad de reestructurar el sitio web mediante acciones *arrastrar y soltar*, sin que se rompa ningún enlace.

2.3.2. Gestión de Contenidos

Una vez que un documento ha sido creado, es almacenado en una base de datos o repositorio que centraliza toda la información del CMS. Se almacena todo el contenido del sitio web junto con otra información relativa a los documentos: Autor, fecha de creación, versión, caducidad.

El repositorio central proporciona al CMS un amplio abanico de características útiles, como son:

- **Control de versiones.** Permite hacer un seguimiento de todas las versiones de una página: Qué ha cambiado, cuándo y por quién fue

cambiado.

- **Control de cambios**. Cada usuario sólo puede cambiar la sección de la que es responsable.
- **Integración**. Integración con las fuentes y tecnologías de la información existentes.

Y lo que es más importante, los CMS tienen la capacidad de funcionar usando flujos de trabajo en los que intervienen usuarios con distintas responsabilidades como pueden ser editores, autores y usuarios.

Por ejemplo, cuando un autor crea una página, ésta puede ser enviada automáticamente a su jefe para su aprobación y después al equipo central encargado del sitio web para su revisión. Finalmente se puede configurar que se envíe al equipo legal para su visto bueno, antes de ser publicado automáticamente en el sitio web. En cada etapa, el CMS gestiona el estado de la página, notificando su situación al personal involucrado en ésta.

De esta forma, la capacidad para funcionar usando flujos de trabajo permite un aumento de los autores involucrados en la gestión del sitio, mientras que se mantiene un estricto control de calidad, precisión y consistencia de la información.

2.3.3. Publicación

Una vez que los contenidos están en el repositorio, se pueden publicar tanto en el sitio web como en una Intranet. La publicación puede ser inmediata o también programada; los contenidos permanecerán accesibles desde su fecha de publicación hasta fecha de caducidad establecida.

Los CMS cuentan con poderosos motores de publicación que permiten aplicar plantillas a los contenidos de manera que el diseño de una página y su apariencia se aplican automáticamente durante su publicación. También ofrecen la posibilidad de que el mismo contenido se publique a múltiples sitios. Este hecho posibilita que los autores puedan centrarse en la redacción del contenido, dejando de lado los problemas de apariencia que dependerán de la configuración del CMS.

2.3.4. Presentación

Dado que cada sitio tiene un aspecto diferente, los CMS permiten a los diseñadores gráficos y desarrolladores web especificar el aspecto que se aplica al sistema. Estas capacidades de presentación asegura que las páginas son consistentes en todo el sitio, y posibilitan una apariencia homogénea.

Existe una separación total entre el contenido y su apariencia visual, de manera que es posible modificar la apariencia, mediante la configuración de las llamadas "templates" o plantillas visuales, sin afectar a los documentos existentes. Además, el CMS gestiona otros muchos aspectos como la compatibilidad con distintos navegadores web y adaptación al idioma, entre los más importantes.

2.4. Los CMS en el Contexto de las Tecnologías de Gestión de Información

Real Story Group[1] es una empresa que lleva más de 10 años trabajando en el campo del análisis de tecnología de contenidos. Previamente era conocida como *CMS-Watch*, pero cambió de nombre en Febrero de 2010. La compañía publica informes de diversas áreas donde el centro de atención para ellos es la tecnología, y su objetivo principal es estudiar cómo se adapta a diversos escenarios. Siguiendo la estructura que sigue su página web para organizar los distintos informes que ofrece la compañía como productos, los diferentes ámbitos o niveles para los proyectos de gestión de información en las organizaciones son:

- **Gestores de Contenidos Web o CMS**. Dentro del ámbito de los CMS se encuentran los WCMS o WCM. Se trata de software que ha sido diseñado para ayudar a las organizaciones a crear y publicar contenidos web y a gestionar los sitios web. Estos sistemas están proliferando mucho en los últimos años debido, sobre todo, a la naturaleza de la Web y a las facilidades de publicación de contenidos que ofrece.

  Los CMS responden a la necesidad de empresas y organizaciones de crear proyectos de comunicación específicos con el objetivo de compartir con otras subsedes y de publicar de cara al público. Están enfocados hacia proyectos web de tamaño mediano, donde existe una amplia comunidad de usuarios creativa y participativa.

  Como CMS destacan Joomla[2], Drupal[3], Ezpublish[4] u Opencms[5].

- **Gestores de Activos Digitales y Multimedia**. Software que se usa cuando existe necesidad de gestionar grandes cantidades de información multimedia (audio, vídeo, imágenes, gráficos, animaciones, juegos, documentos CAD y presentaciones). Es una forma de proporcionar un

[1] `http://www.realstorygroup.com`
[2] `http://www.joomla.org`
[3] `http://drupal.org`
[4] `http://ez.no`
[5] `http://www.opencms.org`

repositorio que facilite la creación, gestión, organización y distribución de ficheros multimedia que se identifican como bienes digitales. Ejemplos de estos sistemas son ADAM Software[6], MediaBeacon[7] y KIT Digital[8].

- **Ecosistemas Sharepoint**. Plataforma web de trabajo colaborativo y de gestión documental, orientada específicamente a documentos ofimáticos. Se integra totalmente con aplicaciones ofimáticas de escritorio en tareas de creación, edición y publicación de documentos. Destacan SurfRay[9] y Microsoft Sharepoint[10].

- **Gestores Documentales o ECMS (*Enterprise Content Management System*)**. Son los sistemas que se centran más en el ciclo de vida de los contenidos de la empresa. Estos sistemas trabajan con diferentes tipos de documentos electrónicos y son capaces de gestionar millones de ellos en cualquier momento. Los sistemas ECM son similares a los CMS en el sentido de que permiten la publicación y difusión de contenidos en la Web o en intranet, aunque dan prioridad al desarrollo de la actividad o negocio corporativo. Destacan Alfresco[11], Documentum[12], OpenText[13] y Nuxeo[14] entre otros.

- **Portales e Integración de Contenidos**. Software que se usa para integrar la información, los datos, las aplicaciones y los procesos en un único punto de acceso web. Se centran en obtener una presentación y autenticación del usuario única desde la cual se interaccione con sistemas y aplicaciones heterogéneas subyacentes, lo que permite también enfocarlos como portales de publicación. Destacan Liferay[15] o Jboss[16], se trata de sistemas basados en arquitectura orientada a servicios web, y uso intensivo de estándares de intercambio de información.

- **Software Colaborativo y Social**. Plataformas, servicios y software mediante el cual una comunidad de individuos trabaja conjuntamente siguiendo un propósito común para alcanzar algún tipo de beneficio. Entre las plataformas más conocidas destacan las de Google y Microsoft; en el ámbito de suite de aplicaciones se puede citar a Awareness[17]

[6] `http://www.adamsoftware.net`
[7] `http://www.mediabeacon.com`
[8] `http://www.kit-digital.com`
[9] `http://www.surfray.com`
[10] `http://sharepoint.microsoft.com`
[11] `http://www.alfresco.com`
[12] `http://spain.emc.com`
[13] `http://www.opentext.com`
[14] `http://www.nuxeo.com`
[15] `http://www.liferay.com`
[16] `http://www.jboss.org`
[17] `http://www.awarenessnetworks.com`

y Traction Software[18]; en cuanto a software a partir del cual desarrollar wikis o blogs, los más conocidos son MediaWiki[19] y Wordpress[20] respectivamente; por último destacan Facebook[21] y Twitter[22] en cuanto a servicios públicos en la Web donde acceder e interactuar con las mencionadas redes sociales.

- **Software de Búsqueda Empresarial**. Los motores de búsqueda empresarial son sistemas que proporcionan a las organizaciones la capacidad de realizar búsquedas seguras de información por medio de sus repositorios de datos. Destaca la plataforma Autonomy[23] y en la categoría de software especializado Sinequa[24] y Vivisimo[25].

A continuación, en la figura 2.2, se puede observar un mapa de vendedores de tecnología de contenidos correspondiente al año 2012 en el que se representan las categorías anteriores a modo de líneas de metro. En el mapa queda representado, dependiendo de su posición, qué productos ocupan una posición central dentro de cada categoría y cómo se relacionan unos productos con otros, indicado a través de la presencia de intercambiadores.

[18] `http://traction.tractionsoftware.com`
[19] `http://www.mediawiki.org`
[20] `http://es.wordpress.com`
[21] `http://www.facebook.com`
[22] `http://www.twitter.com`
[23] `http://www.autonomy.com`
[24] `http://www.sinequa.com`
[25] `http://vivisimo.com`

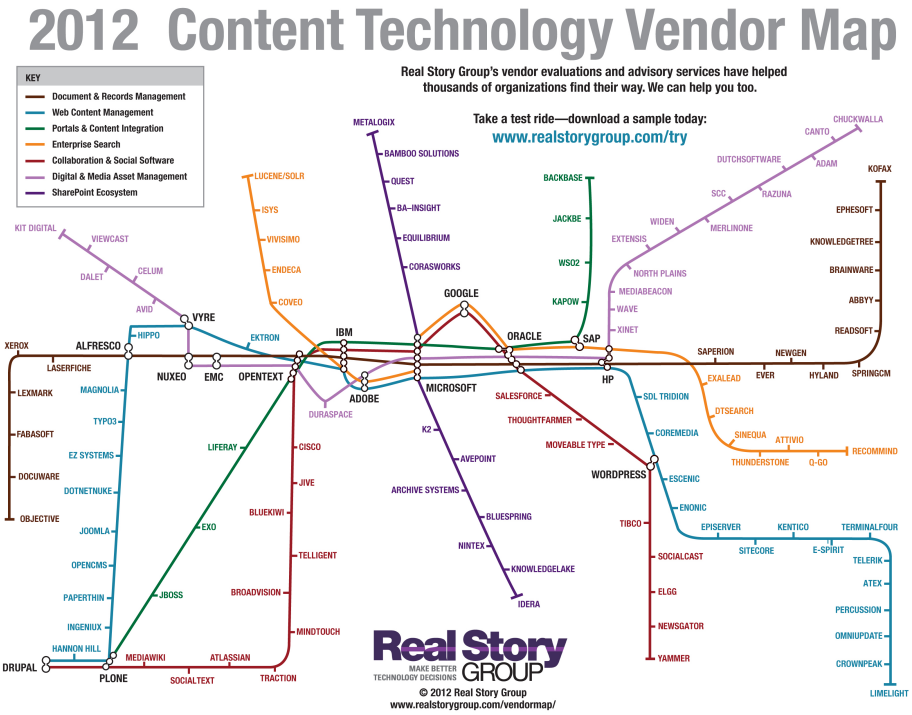


Figura 2.2: Mapa de vendedores de tecnología de contenidos en 2012. Imagen extraída de `http://www.realstorygroup.com/vendormap`.

2.5. Clasificación de los WCMS

Dentro del ámbito de los WCMS, la empresa Real Story Group diferencia en su página web entre los productos comerciales según su penetración en el mercado y orientación hacia diversos tipos de corporaciones. La clasificación de WCMS que propone es la siguiente:

- **Plataformas Empresariales Complejas**. Plataformas que se comercializan a gran escala y que por lo general incluyen paquetes multidimensionales que abarcan muchas funciones; aunque pueden no ser muy adecuados para proyectos sencillos en los que se necesite de las funcionalidades de un WCMS. Ejemplos de estas plataformas son IBM Web Content Manager[26] y Oracle Webcenter Content[27].

- **Plataformas de Alta Gama**. Ocupan un espacio de expansión moderada entre instalaciones de departamentos y vendedores a nivel Empresarial. Ejemplos de estas plataformas son Adobe[28], CoreMedia[29], Percussion[30] y Sitecore[31].

- **Plataformas de Media Gama**. Ofrecen servicios a empresas medianas o proyectos departamentales dentro de grandes empresas. En particular, se centran en escenarios que necesitan una personalización por encima del promedio y que a menudo requieren un consultor externo. Ejemplos de estas plataformas son Drupal[32], Plone[33] y TYPO3[34].

- **Productos de Media Gama**. Productos que ofrecen paquetes de herramientas con características preestablecidas. Muchos de ellos promueven soluciones que incluyen herramientas para trabajar con redes sociales, análisis de sitios y entornos colaborativos, entre otras. Ejemplos de estos productos son Clickability[35], eZ Systems[36] y Magnolia[37].

- **Productos Simples**. Se trata de productos más pequeños, de código libre y/o precios más bajos. Aunque también ofrecen paquetes con

[26] http://www-01.ibm.com/software/lotus/products/webcontentmanagement
[27] http://www.oracle.com/us/products/middleware/content-management/web-content-mgmt
[28] http://www.businesscatalyst.com
[29] http://www.coremedia.com
[30] http://www.percussion.com
[31] http://www.sitecore.net
[32] http://drupal.org
[33] http://plone.org
[34] http://typo3.com
[35] http://cms.clickability.com
[36] http://ez.no
[37] http://www.magnolia-cms.com

muchas herramientas, suelen ser menos ricas o no tan bien probadas como sus principales competidores de niveles superiores. Ejemplos de estos productos son Joomla Project[38], OpenCms[39] y WordPress[40].

2.6. Joomla

Cuando comenzamos la tesis, allá por el año 2008, contábamos con la idea principal de enriquecer los sistemas de gestión de contenidos con las tecnologías de la Web Semántica, para obtener así su equivalente semántico, los S-CMS, y que se beneficiarán así de las bondades que ésta proporciona.

Para empezar necesitábamos elegir un CMS libre de amplia difusión que contase con facilidades de extensibilidad a partir de las cuales realizar desarrollos que ilustrasen nuestras aportaciones en el campo de la Web Semántica. En aquel momento se realizó un estudio de los productos software existentes que cumplieran con este requisito que concluyó con dos candidatos: Joomla y Drupal. Cualquiera de los dos candidatos eran válidos para la labor que se necesitaba. Finalmente nos decantamos por Joomla debido a su estabilidad, difusión y a que contaba con una gran comunidad de desarrolladores de extensiones a la cual podríamos ofrecer nuestras aportaciones.

En esta sección se presenta Joomla, uno de los sistemas de gestión de contenido web libres más extendidos en la comunidad web sobre el que se van a centrar los siguientes capítulos del presente trabajado.

2.6.1. Introducción

Joomla (Kramer, 2010) es un sistema de gestión de contenidos, que permite construir tanto sitios web como potentes aplicaciones en línea. Muchos aspectos, como su facilidad de uso y su extensibilidad, han hecho que Joomla se convierta uno de los programas más famosos a la hora de construir un sitio web. Además, Joomla es una solución de código libre que se encuentra disponible gratuitamente para cualquiera.

Los orígenes de Joomla se remontan al año 2000, con los inicios del desarrollo del CMS propietario conocido como **Mambo** por la empresa *Miro Construct Pty Ltd.* La primera versión de Joomla, la 1.0.0, fue publicada el 16 de septiembre de 2005 y se trataba de una evolución paralela mejorada de Mambo 4.5.2.3 combinada con modificaciones de seguridad y corrección de errores de programación. Durante su primer año de lanzamiento, Joomla

[38] `http://www.joomla.org`
[39] `http://www.opencms.org`
[40] `http://wordpress.org`

obtuvo más de 2,5 millones de descargas. Desde entonces hasta hoy día se han realizado más de 21 millones de descargas.

Joomla ganó el premio al mejor CMS de código libre programado en lenguaje PHP organizado por la editorial *Packt* durante los años 2006 y 2007; el resto de años siempre queda entre las primeras posiciones.

El 22 de enero de 2008 se lanzó la versión 1.5, única versión de Joomla hasta el momento con soporte de larga duración (LTS) y que, además, incorpora notables mejoras en el área de seguridad, administración y cumplimiento con estándares W3C.

Joomla puede considerarse como el CMS más popular del mundo, como así lo evidencia su creciente comunidad de más de 200.000 miembros, entre los que se distinguen usuarios, desarrolladores y colaboradores.

Joomla se usa por todo el mundo para construir sitios web de diversas formas y tamaños, como por ejemplo para:

- Sitios web corporativos o portales
- Intranets corporativas o extranets.
- Revistas online, periódicos y publicaciones.
- Comercio electrónico y reservas online.
- Aplicaciones gubernamentales.
- Sitios web de PYMES.
- Sitios web de organizaciones sin ánimo de lucro.
- Sitios web de escuelas e institutos.
- Páginas web personales y familiares.

A continuación se muestran algunos sitios web que usan Joomla:

- MTV Networks Quizilla (Red social) - `http://www.quizilla.com`
- IHOP (Cadena de restaurantes) - `http://www.ihop.com`
- Harvard University (Educación) - `http://gsas.harvard.edu`
- Citibank (Banca) - No accesible publicamente
- The Green Maven (Organización ecologista) - `http://www.greenmaven.com`

- Outdoor Photographer (Revista) - `http://www.outdoorphotographer.com`
- PlayShakespeare.com (Cultura) - `http://www.playshakespeare.com`
- Denso Interiors (Diseño de muebles) - `http://www.sensointeriors.co.za`

En la URL `http://community.joomla.org/showcase` se pueden encontrar muchos más ejemplos de organizaciones y compañías que usan Joomla como infraestructura a partir de la cual construir sus sitios web.

2.6.2. Características

Joomla fue diseñado para ser fácil de instalar y configurar incluso para usuarios no avanzados. Muchas empresas de hosting ofrecen servicios de instalación, creación y puesta en marcha de un sitio web con una sola pulsación de ratón.

A continuación se muestra una lista con las características estándar de Joomla, aunque la verdaderamente importante es su extensibilidad.

- **Administración de Usuarios.** Dispone de un sistema de registro que permite al usuario configurar sus opciones personales. Existen nueve grupos de usuarios con distintos tipos de permisos mediante los cuales se les permite acceder, editar, publicar y administrar.

  La autenticación es una parte importante de la administración de usuarios y además soporta diversos protocolos entre los que se incluye LDAP, OpenID, e incluso Gmail. Esto permite a los usuarios utilizar la información de sus cuentas para agilizar el proceso de registro.

- **Administración Multimedia.** Dispone de un componente llamado *Media Manager* que permite gestionar fácilmente archivos multimedia, organizarlos en carpetas e incluso configurar cualquier tipo de extensión de fichero MIME (*Multipurpose Internet Mail Extensions*) para poder trabajar con ellas. Además está integrado dentro de la herramienta de edición de artículos para permitir que las imágenes y otros archivos multimedia estén disponibles en cualquier momento.

- **Administración de Idiomas.** Proporciona una herramienta para realizar una internacionalización de sus contenidos y la administración de páginas web en diversos idiomas haciendo uso de la codificación UTF8. Por ejemplo, es posible disponer de un sitio web en un idioma y el panel de administración en otro.

- **Administración de Anuncios**. Permite instalar imágenes y animaciones con anuncios (conocidos como banners) en un sitio web gracias a su administrador de anuncios. Se comienza con la creación de un perfil de cliente, después se añaden las campañas y los anuncios que se necesiten.

- **Administración de Contactos**. Cuenta con una herramienta de administración de contactos que ayuda a los usuarios a encontrar a la persona adecuada y su información de contacto. También dispone de formularios de contacto dirigidos tanto a individuos específicos, como a grupos. Además protege las direcciones de correo electrónico de ser capturadas para ser usadas en el envío de correo basura.

- **Encuestas**. Permite la creación de encuestas para la obtención de información sobre sus usuarios.

- **Búsquedas**. Dispone de herramientas de búsqueda que proporcionan la ayuda necesaria para que los usuarios accedan a los elementos más populares. Además proporciona estadísticas de búsqueda para el administrador.

- **Gestión de Enlaces Web.** Dispone de secciones y categorías a partir de los cuales los usuarios pueden clasificar los enlaces a artículos del sitio web.

- **Gestión de Contenidos**. El sistema de artículos simplificado a tres niveles agiliza la organización de su contenido. Permite organizar el contenido de forma personalizada y no necesariamente de la forma en que aparece en el sitio web. El sistema facilita que los usuarios pueden evaluar los artículos, enviarlos por e-mail a un amigo, o automáticamente sean guardados en un archivo PDF. También posibilita a los administradores que puedan archivar contenido en históricos, ocultándolo a los visitantes del sitio.

  Dispone de un editor WYSIWYG que capacita de una creación de contenidos sencilla, incluso para usuarios principiantes, con la posibilidad de combinar texto e imágenes. Una vez que se han creado los artículos, Joomla proporciona una serie de módulos preinstalados que ofrecen diversas funcionalidades, como por ejemplo: Mostrar los artículos más populares, mostrar las últimas novedades, mostrar flashes informativos, mostrar artículos relacionados y otros más.

- **Sindicación y Gestión de Noticias**. Joomla permite que los usuarios puedan suscribirse al sitio web para que sean informados de cuando se producen nuevos contenidos. También es posible integrar canales RSS de otras fuentes en el propio sitio web.

- **Administración de Menús**. El administrador de menús permite tantos menús y opciones de menú como se necesiten. Permite estructurar los menús de forma jerárquica, de forma completamente independiente de la estructura de su contenido. Es posible poner un menú en varios lugares y en con distintos estilos; proporciona menús desplegables laterales, y casi cualquier otro sistema de navegación que se pueda imaginar. También se genera automáticamente una traza de la ubicación donde se encuentra el usuario para ayudarlo a navegar.

- **Administración de Plantillas**. Las plantillas en Joomla son una forma potente de hacer que un sitio web se presente de la manera deseada. Permite utilizar un modelo único para todo el sitio o bien una plantilla independiente para cada sección del sitio. El nivel de control visual es total y permite personalizar cada parte de las páginas web.

- **Sistema de Ayuda Integrado**. Joomla incorpora un sección para ayudar a los usuarios a encontrar lo que necesitan. Además cuenta con varias herramientas de ayuda como son: Un glosario para explicar los términos en lenguaje llano, un comprobador de versión que asegura que se está utilizando la última versión, una herramienta de información del sistema que ayuda a solucionar problemas y, por si algo falla, vínculos a una gran cantidad de recursos en línea para obtener ayuda y soporte.

- **Servicios Web**. Desde Joomla se pueden utilizar llamadas a procedimiento remoto (a través de HTTP y XML) y usar servicios web. También ofrece la posibilidad de integrar XML-RPC con los servicios de Blogger y las API (*Application Programming Interface*, Interfaz de Programación de Aplicaciones) de Joomla.

- **Otras Características**.

  - Dispone de un cargador rápido de páginas compuesto por una caché de páginas, un módulo que cachea a diferentes niveles de granularidad, y el uso de compresión GZIP de las páginas.
  - Dispone de modo depuración e informe de errores que ayuda al administrador a solucionar los problemas.
  - La capa FTP permite operaciones con archivos, como por ejemplo la instalación de extensiones, sin necesidad de conceder permisos de escritura a todas las carpetas y archivos, haciendo así más fácil la labor del administrador y aumentando la seguridad del sitio web.
  - Dispone de un servicio de mensajería interna a partir del cual el administrador puede comunicarse con los usuarios de uno en uno

a través de mensajes privados, o con todos los usuarios del sitio a través del sistema de correo masivo.

2.6.3. Requisitos de Instalación

Entre los requisitos para la instalación de Joomla están: El uso de *Apache o Microsoft IIS* como servidor web, un intérprete de *PHP* instalado como módulo o como CGI del servidor Web y el uso de *MySQL* como sistema gestor de base de datos. En la misma página del proyecto Joomla se puede encontrar información más detallada sobre el proceso de instalación, el manual de instalación se encuentra accesible en la URL siguiente: *http://help.joomla.org*.

2.6.4. Arquitectura Software

La versión 1.5.X de Joomla es la más difundida y la única que actualmente proporciona soporte de larga duración (LTS). Como muestra la figura 2.3, consta de un sistema que se estructura en tres niveles:

- En el **primer nivel** ubicamos la capa de las extensiones (Extension Layer) que se compone de:
  - **Componentes**: Son pequeños programas independientes entre sí que aportan distinta funcionalidad a Joomla. Parte de los componentes son estándar y forman parte del núcleo del sistema Joomla, ya que se encargan de implementar las distintas características que éste ofrece. Los demás componentes se pueden descargar, dependiendo de las necesidades y la funcionalidad que ofrezcan, e instalarlos desde el panel de administración de Joomla.
  - **Plantillas**: Una plantilla Joomla es un paquete de archivos que controlan la presentación de los contenidos dentro de Joomla. Aunque el diseño y construcción inicial de una plantilla es similar a los que se realizarían en un sitio web, no puede considerarse así ya que necesita de la base de datos de Joomla para tomar la apariencia de un sitio web completo.
  - **Módulos**: Los módulos son pequeñas aplicaciones que pueden situarse en cualquier lugar del sitio. En algunos casos trabajan en conjunción con componentes y en otros son fragmentos de código aislados y completos que se usan para mostrar algunos datos de la base de datos, como por ejemplo contenido. Los módulos se utilizan habitualmente para la salida de información pero también

pueden ser formularios para la entrada de datos (como ejemplos, el Modulo de Acceso o las Encuestas).

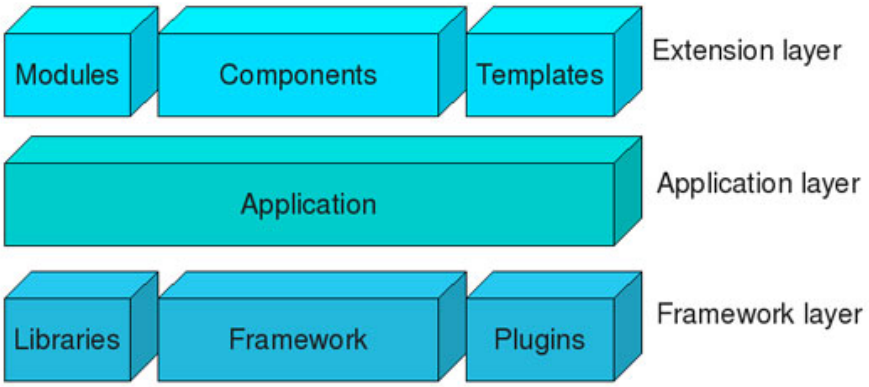


Figura 2.3: Arquitectura software de Joomla. Imagen extraída de `http://docs.joomla.org/Framework/1.5`.

- En el **segundo nivel** tenemos la capa de aplicación (Application layer), compuesta por aplicaciones que extienden la infraestructura software. Actualmente, para la versión 1.5 de Joomla, contiene:
  - **JInstallation**: Responsable del proceso de instalación de Joomla en el servidor web. Esta aplicación debe ser eliminada una vez que se concluye el proceso por motivos de seguridad.
  - **JAdministrator**: Responsable de la administración del sistema.
  - **JSite**: Responsable de administrar la apariencia del sitio de cara a los usuarios finales.
  - **XML-RPC**: Permite la administración remota del sitio web.
- En el **tercer nivel**, tenemos la infraestructura software propiamente dicha, en donde encontramos:
  - **La infraestructura de Joomla**: También se conoce con el término de *Framework*. Se trata de una aplicación reutilizable incompleta que debe de ser especializada para construir aplicaciones completas. La especialización se realiza proporcionando el código necesario para desarrollar la funcionalidad deseada y por tanto para producir la aplicación que se necesite.
  - Las **librerías**: Son bibliotecas de módulos reutilizables que son requeridas por el Framework o son instaladas por terceros (extensiones externas) para atender requerimientos de éstas.

- Los **plugins**: También son los encargados de extender las funcionalidades del Framework aunque de otra manera. Están diseñados para ejecutar partes de código cuando se producen determinados eventos. Son capaces de interactuar con componentes y módulos sin necesidad de modificar su código fuente. Entre las tareas más comunes de los plugin están: Editores de HTML, búsquedas locales, formateado de contenido y el registro de los usuarios.

2.6.5. Programación de Extensiones

Desde el punto de vista del desarrollador y diseñador Web, Joomla posibilita construir sitios web rápidamente. Después se puede instruir a los clientes para que ellos mismos sean responsables de la administración de sus sitios, debido a su facilidad de uso.

En el supuesto de que un cliente en particular necesitase una funcionalidad específica, Joomla es altamente extensible y que existen miles de extensiones, la mayoría gratuitas bajo licencia GPL, disponibles en la siguiente URL: `http://extensions.joomla.org` que ofrece un directorio de extensiones. También ofrece una API desde la cual programar cualquier módulo específico que se necesite a la hora de satisfacer las necesidades de un cliente o sitio web. La API está muy bien documentada en la URL siguiente: `http://api.joomla.org/`

Mientras que la idea de componente parece simple, en la práctica, el código para desarrollarlo puede ser muy complejo conforme se necesiten características adicionales o conforme a la personalización de la interfaz.

MVC (*Model View Controller*, Modelo-Vista-Controlador) (Bergin, 2007) es un patrón de diseño software que se usa para organizar el código de forma que separa la lógica de negocio y la representación de los datos. La premisa detrás de este enfoque es que si la lógica de negocio se agrupa en una sección, entonces la interfaz y la interacción con el usuario que rodea a los datos puede ser revisada y personalizada sin necesidad de reprogramar la lógica de negocio. MVC fue inicialmente desarrollado para mapear los papeles tradicionales de entrada, procesamiento y salida dentro de una arquitectura lógica de interfaz gráfica de usuario.

La manera de relacionarse entre sí de estos tres papeles principales se puede observar en la figura 2.4. Estos papeles son las bases de MVC y se describen a continuación de forma resumida:

- **Modelo**: Es la parte del componente que encapsula los datos de la aplicación. Normalmente proporciona rutinas para manejar y manipular estos datos de forma significativa, además de las rutinas que permi-

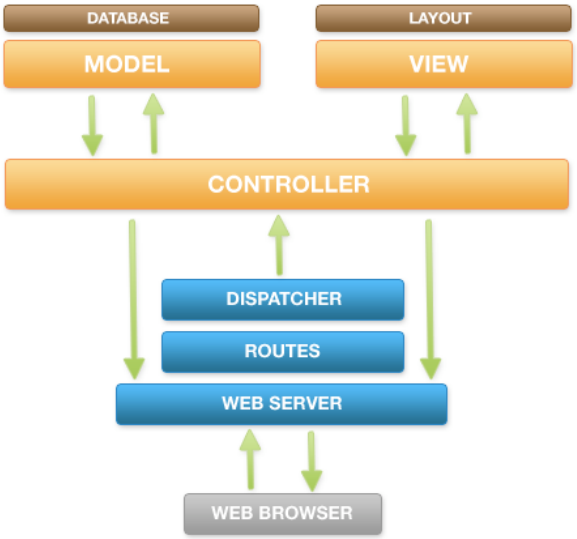


Figura 2.4: Patrón MVC en Joomla.

ten recuperar los datos del modelo. Contiene los métodos para agregar, borrar y actualizar la información almacenada en las tablas de la base de datos. En general, en el modelo se encapsula la técnica de acceso a la base de datos de forma que si una aplicación cambia de gestor de base de datos, el modelo sería el único elemento que se necesitaría cambiar; no sería necesario modificar la vista ni el controlador.

- **Vista**: Se utiliza para representar los datos del modelo de forma que sean susceptibles de interacción. Para una aplicación web, la vista sería una página HTML que devuelve datos. La vista toma los datos del modelo (que se le pasan desde el controlador) y con ellos se cumplimenta una plantilla que se presenta al usuario. La vista no causa que los datos se modifiquen de ninguna manera, sólo muestra los datos recuperados del modelo.

- **Controlador**: Se encarga de responder a las acciones del usuario. En el caso de tratarse de una aplicación web, es una acción del usuario que generalmente corresponde con la solicitud de una página. El controlador determinará qué tipo de solicitud es requerida por el usuario y responderá de forma adecuada mediante la activación del modelo, la manipulación de los datos y el traspaso a la vista. El controlador no muestra los datos del modelo, sólo activa en métodos del modelo que modifican los datos, y luego se pasan a la vista encargada de mostrarlos.

2.7. Áreas Relacionadas

Esta sección realiza una introducción a otras áreas íntimamente relacionadas con los sistemas de gestión de contenidos como son por portales web y la variante semántica tanto de éstos como de los CMS, que se usan para establecer los precedentes contextuales sobre los que se desarrolla la presente tesis.

2.7.1. Portales Web y Portales Web Semánticos

Un "portal web" es un sitio web que funciona como punto de acceso a la información de la Web (Christ et al., 2002). Se trata de un intermediario de información cuyo objetivo es presentar la información de forma unificada y ofrecer servicios tales como motores de búsqueda de información, correo electrónico, noticias, bases de datos y entretenimiento entre otros. Uno de los objetivos más importantes de los portales es el de permitir a empresas y organizaciones proporcionar una apariencia consistente para sus procedimientos de control de acceso web, bases de datos y aplicaciones web que de otra forma serían diferentes. Toda la información que maneja un portal web no tiene porque estar contenida dentro de sí mismo, sino que normalmente, se encuentra distribuida en diversos sitios web y es el portal el que se encarga de centralizar los enlaces de forma organizada dependiendo de la complejidad y heterogeneidad de la misma.

"Portal web semántico" (Fernández-García et al., 2006) es un portal web, tanto particular como empresarial, que utiliza internamente las tecnologías de la web semántica. Los usuarios interactúan con el portal como si de un portal web tradicional se tratara, pero además de mostrarse información mediante páginas web estáticas o dinámicas, se genera información semántica mediante los lenguajes de marcas RDF o RDFa que puede ser accesible a agentes software semánticos. El objetivo de añadir contenidos semánticos es ayudar en la tareas de búsqueda, filtrado y explotación de la información recogida en dichos portales.

2.7.2. Sistemas Semánticos de Gestión de Contenido

Se trata de sistemas que, a día de hoy, están emergiendo y no existe ninguna definición aceptada de momento.

Teniendo en cuenta todo lo expuesto en esta sección, extrapolándolo, se entiende por S-CMS el sistema donde se integran las herramientas propias de los CMS con herramientas semánticas (Bratsas et al., 2012). Se trata de sistemas que disponen de herramientas que posibilitan a usuarios sin cono-

cimientos técnicos gestionar un sitio web de forma sencilla, sin descuidar la semántica de los contenidos que manejan. Además de las herramientas propias de los CMS, contarán con herramientas de anotación semántica que ayudarán a enriquecer los contenidos con metadatos de forma transparente para el usuario. También se dispondrá de herramientas de explotación, como por ejemplo herramientas de inferencia o de búsqueda basada en la información semántica que recogen dichos metadatos.

En definitiva, al igual que ocurre en la subsección anterior con los portales web semánticos, un S-CMS es un sistema de gestión de contenidos en donde además de aportar la funcionalidad asociada a los CMS, se dispone de nueva funcionalidad para enriquecer los contenidos de los mismos con metadatos que se incorporan a los documentos mediante lenguajes de marcas tipo RDF o RDFa, y permiten beneficiarse de las funcionalidades que ofrece la Web Semántica en tareas como la búsqueda mejorada de información, la inferencia de nueva información a partir de una base y la mejora general en la explotación de la información recogida.

2.8. Conclusiones

En la actualidad, muchas empresas que implementaban webs han cerrado o han tenido que adaptarse al hecho de que los CMS se han convertido en un artículo de consumo. Existen multitud de distribuciones de gestores de contenido tanto de código libre como comerciales. Por consiguiente, los campos en los que más se está trabajando en el sector de los CMS son la creación de extensiones para la ampliación de sus funcionalidades y la creación de plantillas visuales o templates para la personalización de la apariencia de los mismos.

Por otro lado, los CMS han ido incorporado los estándares propuestos por la W3C, entre otros se pueden citar XML, XHTML, CSS y RSS, que han aportado calidad y estabilidad a los sistemas.

Con respecto al futuro, es previsible que las tendencias continúen y sea necesario establecer unos estándares de almacenamiento, estructuración y gestión de contenido con objeto de conseguir mayor estabilidad e incluso compatibilidad entre sistemas web de gestión de contenidos con similar funcionalidad.

También parece próxima la fusión de los diferentes tipos de CMS, en particular la fusión los ECMS con los CMS. La razón es que actualmente se están perfeccionando y difundiendo distintos entornos colaborativos para trabajar con documentos ofimáticos de forma online. Prueba de esto son por

ejemplo "Google Docs"[41] y "Office Web Apps"[42]. La fusión será posible en la medida en que estas herramientas consigan suficiente madurez y las APIs que ofrecen la funcionalidad para trabajar con este tipo de documentos estén accesibles a la comunidad.

Desde el punto de vista de la presente tesis, es de sumo interés que se produzca una adopción paulatina de los estándares propuestos por la W3C, tales como RDF, RDFa y OWL, para enriquecer con metadatos los documentos web.

En este capítulo se han estudiado los problemas que se presentan en el desarrollo de sitios web y se han presentado los CMS como sistemas que facilitan la creación y gestión de sitios web. Se ha estudiado la funcionalidad que ofrecen, los distintos tipos de CMS que existen acorde con el contexto en el que se usan y el papel que juegan dentro de las tecnologías de gestión de la información. También se ha estudiado Joomla, el CMS sobre el que se centrarán los estudios y desarrollos de los siguientes capítulos de la presente tesis. Por último se ha realizado una introducción a áreas relacionadas con los CMS como son los portales web semánticos y los sistemas semánticos de gestión de contenidos, estableciendo las bases para las propuestas que se realizan en los siguientes capítulos de la tesis.

[41] `http://office.live.com/`
[42] `http://docs.google.com`

Capítulo 3

La Web Semántica: Principios y Situación

Nunca considere el estudio como un deber, sino como una oportunidad para penetrar en el maravilloso mundo del saber.

Albert Einstein

Resumen: En este capítulo se hace un estudio de la Web Semántica, un área prolífera cuyas posibilidades aún no se han explotado del todo. Se presentan los conceptos clave, su modelo constructivo y las tecnologías necesarias para llevarlo a la práctica.

3.1. Introduccion

Según la definición de la W3C[1]: *"La Web Semántica es una Web extendida, dotada de mayor significado, en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Al dotar a la Web de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de una infraestructura común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla. Esta Web extendida y basada*

[1] `http://www.w3c.es/divulgacion/guiasbreves/websemantica`

en el significado, se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante".

La Web ha cambiado profundamente la forma en la que nos comunicamos, hacemos negocios y realizamos nuestro trabajo. La comunicación prácticamente con todo el mundo en cualquier momento y a bajo coste es posible hoy en día. Podemos realizar transacciones económicas a través de Internet. Tenemos acceso a millones de recursos, independientemente de nuestra situación geográfica e idioma. Todos estos factores han contribuido al éxito de la Web. Sin embargo, al mismo tiempo, estos factores que han propiciado el éxito de la Web, también han originado sus principales problemas: Sobrecarga de información y heterogeneidad de fuentes de información con el consiguiente problema de interoperabilidad.

La Web Semántica ayuda a resolver estos dos importantes problemas permitiendo a los usuarios delegar tareas en software. Gracias a la semántica en la Web, el software es capaz de procesar su contenido, razonar con éste, combinarlo y realizar deducciones lógicas para resolver problemas cotidianos automáticamente. La Web Semántica es la dirección principal del futuro del desarrollo Web. Como se afirma en Berners-Lee et al. (2001), es *"una extensión de la actual Web donde la información es proporcionada con un significado bien definido, permitiendo a computadores y personas cooperar".*

Según otros autores (Perojo y León, 2005), la Web Semántica es una extensión de la Web cuya idea básica es proporcionar una infraestructura que permita que las páginas Web, las bases de datos, los programas y aplicaciones, los dispositivos, tanto personales como los empleados en el hogar, puedan consumir y producir datos, sin los problemas causados por los diferentes protocolos de acceso a la información que hacen de la transferencia de contenidos una tarea ardua y difícil. La Web Semántica es un área prolífera, situada en la confluencia de la inteligencia artificial y las tecnologías Web, que propone nuevas técnicas y paradigmas para la representación de la información y el conocimiento; para facilitar, tanto localizar como compartir, integrar y recuperar recursos.

Dicho enfoque propone enriquecer la estructura de la información y agregar componentes semánticos que puedan procesarse de forma automática. La nueva generación de formatos está encabezada por XML (*eXtensible Markup Languaje*, Lenguaje de Marcas Extensible) y RDF (*Resource Description Framework*, Marco de Descripción de Recursos), los cuales incluirán ontologías (taxonomías de conceptos con atributos y relaciones que proporcionan un vocabulario consensuado para definir redes semánticas de unidades de información interrelacionadas) que especificarán las reglas lógicas para que los agentes de software reconozcan y clasifiquen cada concepto (Castells et

al., 2003).

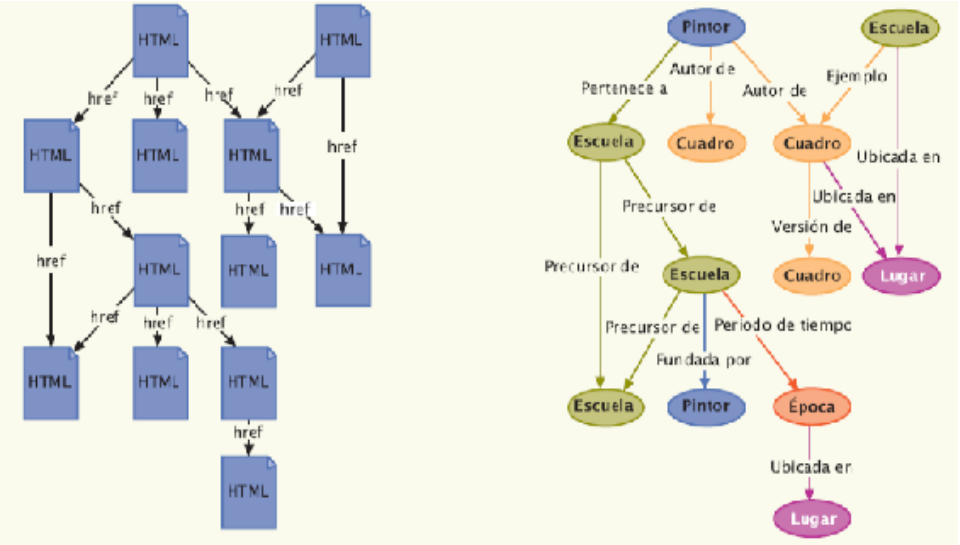


Figura 3.1: Web sintáctica actual vs. Web Semántica.

La Figura 3.1 ilustra esta propuesta (Fonseca et al., 2006). Actualmente la Web se asemeja a un grafo formado por nodos del mismo tipo, y arcos que representan hiperenlaces igualmente indiferenciados. Se puede decir, por ejemplo, que no se hace distinción entre la página personal de un pintor y el portal de una galería de arte online, como tampoco se distinguen explícitamente los enlaces a las páginas personales de los pintores de los enlaces a fotografías de sus cuadros. Por el contrario, en la Web Semántica cada nodo (recurso) tiene un tipo (pintor, escuela, cuadro, lugar, etc.), y los arcos representan relaciones explícitamente diferenciadas (pintor-escuela, pintor-cuadro, escuela-lugar).

La evolución de la Web, en opinión de Pablo Castells (Castells et al., 2003), durante los últimos años, no puede pasar por alto los siguientes acontecimientos:

- **1989:** Tim Berners Lee presenta su proyecto WWW (*World Wide Web*) en el CERN (*Conseil Européen pour la Recherche Nucléaire*).
- **1993:** Creación de los primeros servidores Web y el navegador Mosaic.
- **1994:** Creación del W3C (*World Wide Web Consortium*).
- **1997:** Creación de SHOE (*Simple HTML Ontology Extensions*), primer antecedente de la Web Semántica, basado en HTML.

3.2. Arquitectura de la Web Semántica

En cuanto a las tecnologías y lenguajes necesarios para la implementación de la Web semántica, se puede esquematizar de forma gráfica como la infraestructura que se muestra en la figura 3.2, dividida en varias capas o niveles, donde cada capa explota y utiliza las capacidades de capas inferiores. Este diagrama, presentado por Berners-Lee en la XML Conference del año

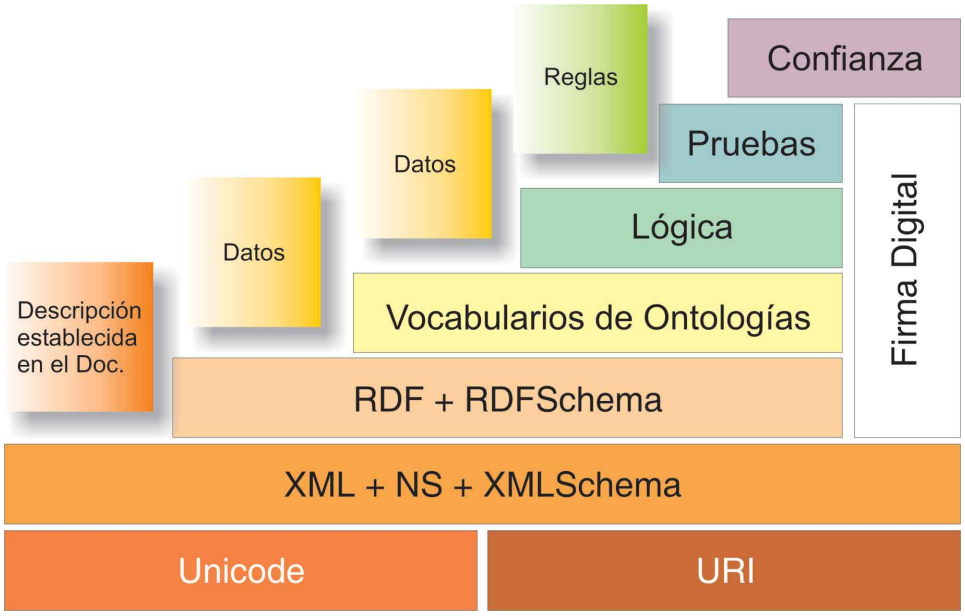


Figura 3.2: Modelo de capas propuesto por Berners-Lee para la Web Semántica. Imagen extraída de `http://http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html`.

2000, sirve de aproximación visual al conjunto de tecnologías que forman el esquema de capas mencionado, cuyos componentes son:

- En la base del modelo de capas, se pueden observar dos elementos elementales: La codificación UNICODE y las cadenas tipo URI (*Uniform Resource Identifier*, Identificador Uniforme de Recursos). La codificación de texto UNICODE permite utilizar caracteres y alfabetos internacionales, ofreciendo la posibilidad de tener información en la Web Semántica expresada en cualquier idioma. Las URIs permiten la definición y localización de recursos de forma unívoca.
- En la segunda capa se encuentran los lenguajes XML y XML Schema y la recomendación NS (*NameSpaces*, Espacios de Nombres). El lenguaje XML se utiliza como base sintáctica para la estructuración del

contenido en la Web y ofrece un formato común para el intercambio de documentos. XML Schema es un lenguaje para describir la estructura de un documento XML y restringir su contenido. La recomendación NS es un método para proporcionar elementos y atributos con nombre único en un documento XML, permitiendo asociar con precisión cada propiedad con el vocabulario XML en el que se define dicha propiedad.

- En la tercera capa se encuentran los lenguajes RDF y RDFS (*RDF Schema*). RDF actúa como modelo básico para establecer propiedades sobre los recursos, ya que permite describir recursos mediante tripletas sujeto-predicado-objeto. RDFS es una extensión semántica de RDF que proporciona los elementos básicos para la descripción de vocabularios; se usa como modelo para definir relaciones entre los recursos por medio de clases y objetos, que se expresan mediante sus esquemas.

- En la cuarta capa se encuentran lenguajes para la representación de ontologías, como OWL (*Web Ontology Language*, Lenguaje de Ontologías Web), que ofrecen un modelo para catalogar y clasificar la información. Además, estos lenguajes hacen posible la interoperabilidad y reutilización entre ontologías de diversos dominios del conocimiento en la Web.

- Una capa lógica que permita realizar consultas y donde se establecen reglas de inferencia que puedan utilizarse para razonar sobre los datos consultados. Se trata de una capa donde, apoyándose en las capas anteriores, es posible conseguir la interoperabilidad entre aplicaciones y sistemas de información heterogéneos.

- En las capas siguientes se encuentran las capas de "pruebas" y de "confianza" que interaccionan entre sí para comprobar de manera cuidadosa las fuentes de la información. Se encargan de evaluar las reglas de la capa lógica y determinar la confiabilidad de un recurso.

- En vertical, de forma trasversal a varias capas, nos encontramos con la firma digital, una herramienta criptográfica de gran utilidad a la hora de verificar que los intercambios de información se realizan de forma segura y que la información ha sido ofrecida por una fuente de confianza.

La pila de capas se encuentra todavía en evolución y poco a poco se han ido concretando las tecnologías de las diversas capas para dar lugar la arquitectura actual que es la que se muestra a continuación en la figura 3.3. En la figura 3.3 aparecen algunas diferencias con respecto a la arquitectura presentada en la figura 3.2. En general, dichas diferencias son debidas a la concreción de las tecnologías que intervienen en la capa. Cabe destacar las siguientes:

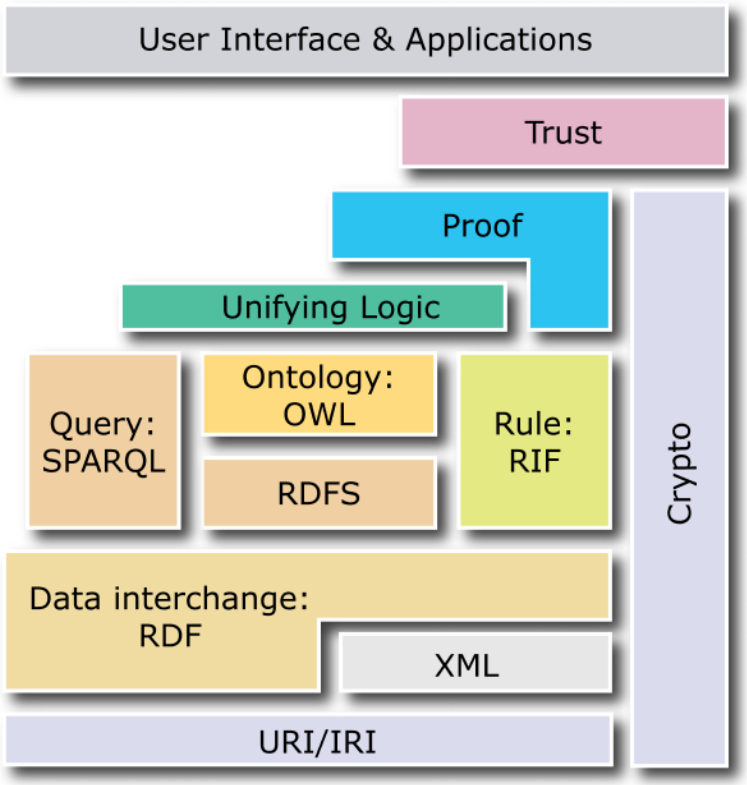


Figura 3.3: Modelo de capas de la Web Semántica actual. Imagen extraída de `http://www.w3.org/2009/Talks/0120-campus-party-tbl/`.

- En la base de la pila de capas aparece un nuevo elemento de protocolo llamado IRI (*Internationalized Resource Identifiers*, Identificador de Recursos Internacionales), encargado de delimitar un recurso en un determinado idioma.

- En cuanto al intercambio de datos, XML continúa siendo el lenguaje referencia para representación e intercambio de información estructurada en la Web y RDF, expresado en sintaxis RDF/XML, se sigue usando como modelo para la representación de las propiedades de los recursos.

- En la nueva arquitectura aparecen explícitamente RDFS y OWL como lenguajes para la representación del conocimiento mediante ontologías.

- Aparece un nuevo estándar llamado RIF (*Rule Interchange Format*, Formato de Intercambio de Reglas), se trata de un lenguaje en XML cuyo objetivo es construir sistemas de reglas en la Web. Las reglas

declarativas permiten la integración y la transformación de datos provenientes de múltiples fuentes de forma distribuida, transparente y escalable.

- Aparece otro lenguaje estándar para la consulta de grafos RDF, llamado SPARQL (*SPARQL Protocol and RDF Query Language*, Protocolo SPARQL y Lenguaje de Consulta RDF). Supera las limitaciones de los lenguajes de consulta tradicionales de búsquedas locales y formatos sencillos. Se encarga de ofrecer primitivas de consulta a la capa lógica a partir de las cuales realizar inferencias.
- Por último, se puede observar que el término "Firma Digital" que aparecía en la figura 3.3 ha sido sustituido por "Criptografía", un término más general que abarca distintos tipos de algoritmos tanto de llave pública como privada.

3.3. Tecnologías de la Web Semántica

A continuación se realiza una presentación de los conceptos citados anteriormente en la sección 3.2, así como otras tecnologías implicadas.

3.3.1. URL / URI / IRI

Dada la necesidad de definir recursos de forma unívoca, se necesitan estándares para la localización de recursos de información en la Web de forma inequívoca y única, así como para la codificación de caracteres a nivel internacional.

Una dirección URL (*Uniform Resource Locator*, Localizador Uniforme de Recursos) (Berners-Lee et al., 1994) es un identificador que permite acceder a recursos. Los URL son usados por los diversos protocolos de Internet (HTTP, FTP, POP, SMTP, SSH, TELNET y otros) para encontrar una página o recurso Web en el vasto mundo del Internet.

Una dirección URI (Berners-Lee et al., 2005) se diferencia de una URL en que permite incluir en la dirección una subdirección, conocida como fragmento, dentro del recurso al que referencia la dirección. El fragmento está delimitado por el carácter numeral "#" y se extiende hasta donde se termina el URI.

Una dirección IRI (Duerst y Suignard, 2005) es una secuencia de caracteres de UNICODE (Consortium, 1991) que sirve para delimitar un recurso en un determinado idioma. La especificación IRI proporciona la referencia definitiva para identificadores que soportan caracteres internacionales. De

acuerdo a la especificación IRI, cada URI ya es un IRI. El objetivo es conseguir que desarrolladores y usuarios puedan identificar recursos en su propio idioma.

```
1 <protocolo>://<parte jerárquica>?<solicitud>
2 <protocolo>://<parte jerárquica>?<solicitud>#<fragmento>
3 http://www.dominio.es/ruta/documento.html?variable1=valor1&
      variable2=valor2
4 http://www.dominio.es/ruta/documento.html#subdireccion
```

Ejemplo 3.1: Formato de URL y URI.

En las líneas 1 y 2 del ejemplo 3.1 se puede observar los formatos para la generación de URLs y URIs respectivamente. En las líneas 3 y 4 se puede encontrar ejemplos de URL y URI. Algunas aclaraciones al respecto:

- La etiqueta denominada *"<protocolo>"* identifica el protocolo de Internet que se va a usar. El más común es sin duda el *http*, aunque existe una gran variedad: Fax, ftp, file, gopher, http, https, imap, ldap, mailto, news, nfs, nntp, pop, sip, sips, snmp y telnet entre otros. En las líneas 3 y 4 del ejemplo 3.1 correspondería con el protocolo *http*.
- La etiqueta denominada *"<parte jerárquica>'* contiene la información del dominio o dirección IP para acceder al servidor y la ruta en el servidor para acceder al recurso. En las líneas 3 y 4 del ejemplo 3.1 correspondería con la ruta "www.dominio.es/ruta/documento.html".
- La etiqueta denominada *"<solicitud>"* indica variables (GET) que se pasan al recurso (en el caso de tratarse de páginas Web dinámicas). Está separada de la parte jerárquica mediante el signo "?" y termina donde empieza el fragmento delimitado por el carácter "#". En la línea 3 del ejemplo 3.1 correspondería con la cadena "documento.html?variable1=valor1&variable2=valor2".
- Por último, la etiqueta denominada *"<fragmento>"* permite indicar una subdirección dentro del recurso al que apunta la dirección. En la línea 4 del ejemplo 3.1 correspondería con la cadena siguiente: "Documento.html#subdireccion".

3.3.2. XML

XML (Bray et al., 2008) nace en el año 1996, fue ratificada por el World Wide Web Consorsium en 1998 mediante la especificación XML 1.0 con el fin de posibilitar el intercambio de documentos estructurados por medio de la Web, a la vez que permitía el uso de hipertexto.

Su potencia proviene de la separación que ofrece entre la interfaz de usuario y la estructura de los datos, se centra en la definición de los contenidos. Se separan los datos de la representación y del procesamiento, así, permite mostrar y procesar los datos como se desee, en dependencia de las diferentes aplicaciones u hojas de estilo empleadas.

Uno de los aspectos más importantes del XML es que es un conjunto de tecnologías basadas en estándares abiertos, que forman módulos opcionales y que amplían sus posibilidades (Perojo y León, 2005). Algunos de estos módulos son:

- `DTD (`*`Document Type Definition`*`)`: Conjunto formal de declaraciones de elementos, atributos y entidades que le indican a un sistema exactamente el tipo de etiquetado que se utiliza en dicho documento.

```
<!DOCTYPE etiqueta[
<!ELEMENT etiqueta (nombre,calle,ciudad,pais,codigo)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT calle (#PCDATA)>
<!ELEMENT ciudad (#PCDATA)>
<!ELEMENT pais (#PCDATA)>
<!ELEMENT codigo (#PCDATA)>]>
```

Ejemplo 3.2: Ejemplo de DTD.

En el ejemplo 3.2 anterior se puede observar la definición de una entidad llamada *"etiqueta"*, que se compone a su vez de cinco propiedades: *Nombre,calle,ciudad, país y código.*

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="etiqueta">
  <xsd:element name="nombre" type="xsd:string"/>
  <xsd:complexType>
  <xsd:sequence>
    <xsd:element name="calle" type="xsd:string"/>
    <xsd:element name="ciudad" type="xsd:string"/>
    <xsd:element name="pais" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attribute name="codigo" type="xsd:double"/>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

Ejemplo 3.3: Ejemplo de esquema XML.

- `XML Schema`: Si bien los DTD permiten describir documentos, un esquema es algo más restrictivo, más similar a un esquema de bases de

datos en que el contenido de los elementos tiene asociado un tipo. Un esquema permitiría a un procesador validar el documento por inconsistencias de una forma más apropiada.

En el ejemplo 3.3 anterior se realiza la misma definición de la entidad del ejemplo 3.2 (etiqueta), expresada en esta ocasión mediante un esquema XML.

```
<etiqueta>
<nombre>Fulano Mengánez</nombre>
<calle>c/ Mayor, 27</calle>
<ciudad>Valderredible</ciudad>
<pais>España</pais>
<codigo>39343</codigo>
</etiqueta>
```

Ejemplo 3.4: Ejemplo de documento XML.

En el ejemplo 3.4 se presenta un documento XML codificado según el DTD y esquema XML de los ejemplos 3.2 y 3.3, donde se puede observar cómo se define una instancia de la entidad *etiqueta* y se asignan valores a sus propiedades.

3.3.3. Espacios de Nombres

XML fue creado con la finalidad de permitir la interoperabilidad sintáctica entre aplicaciones y esquemas de metadatos en el que sus creadores pueden diseñar y mantener sus propios vocabularios en esta sintaxis.

```
<root xmlns:h="http://www.w3.org/TR/html4/"
        xmlns:f="http://www.w3schools.com/furniture">
<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>
<f:table>
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
</root>
```

Ejemplo 3.5: Ejemplo de uso de espacios de nombres XML.

Se puede producir entonces una gran confusión si diferentes desarrolladores escogiesen los mismos nombres de elementos para representar diferentes entidades; por ello, se introdujeron los espacios de nombres, en inglés conocidos como *"namespaces"* (Bray et al., 2009), en XML para resolver este problema, y posibilitar así el uso de múltiples vocabularios en un mismo documento.

Un espacio de nombres se define mediante la propiedad **xmlns** en la etiqueta inicial de un elemento. Su declaración tiene la siguiente sintaxis: "xmlns : prefijo = URI", donde *prefijo* es una etiqueta cualquiera que sirve para referirse a las propiedades que se especifican en el esquema XML de la dirección indicada en la etiqueta *URI*.

En el ejemplo 3.5 anterior, el atributo xmlns en la etiqueta "<table>" dota a los prefijos "h:" y "f:" con un espacio de nombres calificados (en particular, el prefijo "h:" se usará con un esquema XML que define las etiquetas de HTML y el prefijo "f:" un esquema para trabajar con muebles) . Cuando un espacio de nombres es definido para un elemento, todos los elementos secundarios con el mismo prefijo se asocian al mismo espacio de nombres.

3.3.4. RDF

RDF (Manola y Miller, 2004) viene de *"Resource Description Framework"*. Se creó en agosto de 1997, bajo el auspicio del W3C con la finalidad de crear una infraestructura para la descripción de recursos que proporcione una base para procesar metadatos y posibilitar la interoperabilidad semántica entre aplicaciones web.

RDF es un modelo simple para la representación de los metadatos. Permite definir información sobre cualquier dominio. Todo lo que se describen son recursos identificados por URIs (Perojo y León, 2005).

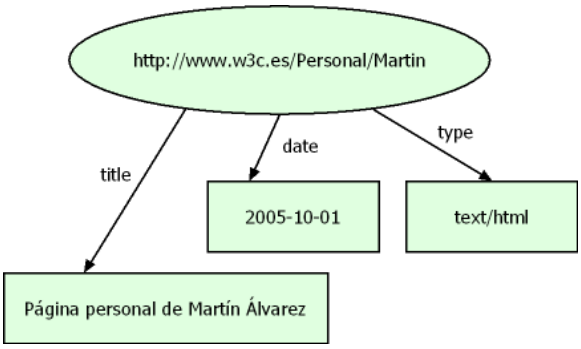


Figura 3.4: Grafo RDF.

El modelo RDF es un grafo definido como una tripleta

- `Sujeto:` Recurso.
- `Predicado:` Propiedad.
- `Objeto:` Literal o recurso.

RDF utiliza XML para la codificación de metadatos. Como todo lo expresable en RDF es expresable en XML, podría surgir la pregunta de por qué es necesario RDF si todo metadato representado en RDF puede también ser representado en XML. La razón es que RDF provee un método estandarizado de representación de metadatos en XML. Usando directamente XML para la representación de metadatos, podrían obtenerse varias representaciones diferentes.

En el ejemplo 3.6 que se muestra a continuación se realiza una traducción al lenguaje RDF de la figura 3.4, usando sintaxis XML.

```
<rdf:RDF>
  <rdf:Description about="http://www.w3c.es/Personal/Martin">
       <s:type>text/html</s:type>
       <s:date>2005-10-01</s:date>
       <s:title>Página personal de Martín Álvarez</s:title>
  </rdf:Description>
</rdf:RDF>
```

Ejemplo 3.6: Representación de la figura 3.4 en RDF/XML.

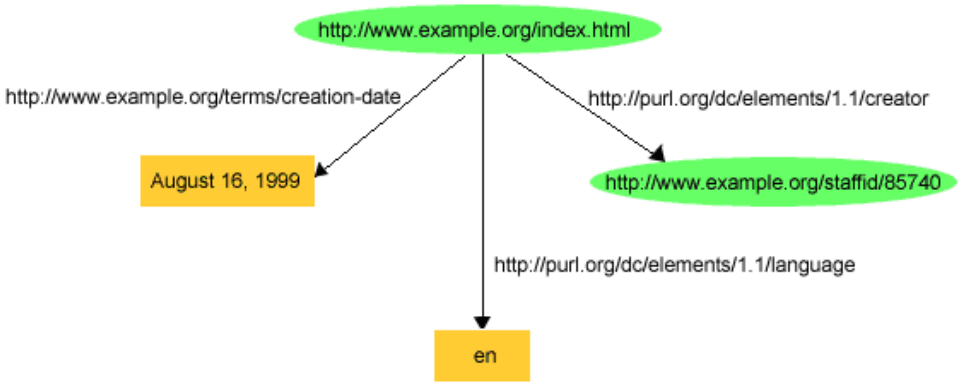


Figura 3.5: Grafo RDF.

La figura 3.5 muestra un grafo RDF en el cual el sujeto que está arriba tiene tres predicados distintos, de los cuales dos de ellos son literales (caja cuadrada) y uno de ellos es un objeto tipo recurso (caja redondeada).

El ejemplo 3.5 muestra la traducción a lenguaje RDF del grafo de la Figura 3.5. Este es un ejemplo donde las tres sentencias RDF que aparecen en el grafo se han agrupado de forma adecuada.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:exterms="http://www.example.org/terms/">
    <rdf:Description rdf:about="http://www.example.org/index.html">
      <exterms:creation-date>August 16, 1999</exterms:creation-date>
      <dc:language>en</dc:language>
      <dc:creator rdf:resource="http://www.ex.org/staffid/85740"/>
    </rdf:Description>
</rdf:RDF>
```

Ejemplo 3.7: Traducción a RDF del grafo de la figura 3.5.

Más adelante, en la sección 4.2.1, se presentará la forma de realizar anotaciones semánticas usando el marco de descripción de recursos RDF.

3.3.5. Esquema RDF

El modelo RDF no facilita por sí solo los mecanismos para la definición de propiedades y relaciones entre predicados y objetos.

Un *"esquema RDF"* o abreviadamente *"RDFS"* (Brickley y Guha, 2004) es un lenguaje de descripción de vocabularios RDF que sirve para hacer explícitas relaciones jerárquicas que se establecen entre ellos, o bien para matizar el carácter obligatorio u opcional de las propiedades y otras restricciones.

RDFS permite modelar metadatos con una representación explícita de su semántica y permite especificar restricciones de tipos de datos para los sujetos y objetos de las tripletas de RDF, introduciendo unas primitivas de modelado orientado a objetos: *rdfs:Class, rdfs:Property, rdfs:subClassOf*. No proporciona vocabularios específicos, sino facilidades para describir las clases y propiedades de un dominio específico.

RDFS facilita la combinación de sentencias RDF, a través de los URIs. También ofrece la base para poder realizar razonamientos o deducciones sobre nueva información.

El papel de RDFS en la terminología de Ingeniería del Conocimiento es definir una ontología simple que documentos RDF particulares puedan chequear, para decidir su consistencia.

RDFS permite definir los términos que se usarán en las declaraciones RDF y les otorgará significados específicos. Para evitar definiciones conflictivas del mismo término, RDF utiliza los espacios de nombres de XML.

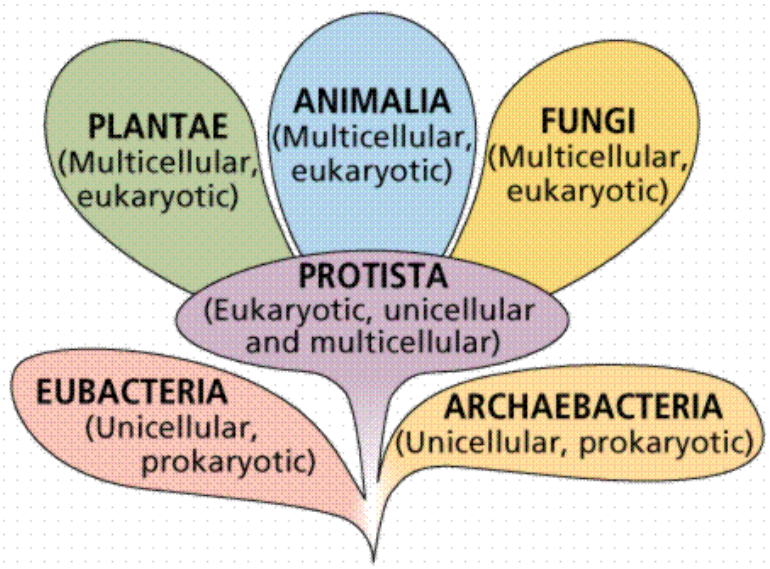


Figura 3.6: Ejemplo de taxonomía. Imagen extraída de `http://www.cartage.org.lb/en/themes/Sciences/Zoology/AboutZoology/DiversityLife/DiversityLife.htm`.

En la figura 3.6 se puede observar un ejemplo de taxonomía donde se representan los diferentes reinos de seres vivos.

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:base="http://ejemplo.es/kingdoms#">
<rdfs:Class rdf:ID="eubacteria" />
<rdfs:Class rdf:ID="archaebacteria" />
<rdfs:Class rdf:ID="protista" />
<rdfs:Class rdf:ID="eukaria" />
<rdfs:Class rdf:ID="plantae">
  <rdfs:subClassOf rdf:resource="#eukaria"/>
</rdfs:Class>
<rdfs:Class rdf:ID="animalia">
  <rdfs:subClassOf rdf:resource="#eukaria"/>
</rdfs:Class>
<rdfs:Class rdf:ID="fungi">
  <rdfs:subClassOf rdf:resource="#eukaria"/>
</rdfs:Class>
</rdf:RDF>
```

Ejemplo 3.8: Representación de la figura 3.6 en RDF/XML.

En el ejemplo 3.8 anterior, se puede observar un esquema RDF donde se representa la taxonomía de la figura 3.6.

3.3.6. Ontologías

El término ontología procede de la Filosofía, en donde se define como *"el estudio del ente en cuanto a tal"* (Smith, 2003). Por ello es llamada la teoría del ser, esto es, el estudio de las cosas: Qué es, cómo es y cómo es posible. Así, la Ontología como ciencia se ocupa de establecer las categorías fundamentales o modos generales de ser de las cosas.

En el ámbito de la Informática, el concepto ha ido variando a lo largo del tiempo dependiendo del uso que se le ha dado en cada momento. En general, el término ontología hace referencia al intento de formular un esquema conceptual riguroso y exhaustivo dentro de un dominio determinado, con la finalidad de facilitar la comunicación y la reutilización de la información entre diferentes sistemas.

La definición más aceptada de ontología es la propuesta por Gruber (1993) y ampliada por Studer et al. (1998), que establece que una ontología es *"una especificación explícita y formal sobre una conceptualización compartida"*. El significado de esta definición es que las ontologías definen los elementos presentes en un dominio de una forma consensuada y accesible utilizando un lenguaje que puede ser procesado por un ordenador.

La explicación de la definición de ontología del autor R. Studer et al. es la siguiente: *"Conceptualización se refiere a un modelo abstracto de algún fenómeno en el mundo a través de la identificación de los conceptos relevantes de dicho fenómeno. Explícita significa que el tipo de conceptos y restricciones usados se definen explícitamente. Formal representa el hecho de que la ontología debería ser entendible por las máquinas. Compartida refleja la noción de que una ontología captura conocimiento consensual, esto es, que no es de un individuo, sino que es aceptado por un grupo"*.

Al revisar la bibliografía especializada, se aprecia que los elementos que forman una ontología son variables dependiendo del dominio de interés y las necesidades de los desarrolladores. A continuación se presentan los elementos de las ontologías que se describen en Noy y McGuinness (2001) y Sowa (2000).

- **Conceptos o clases**. Corresponden con las ideas básicas que se pretenden formalizar y determinan conjuntos de objetos del dominio. Los conceptos se organizan en jerarquías donde un concepto de nivel superior generaliza a otro de nivel inferior.

- **Relaciones**. Se refiere al tipo de asociación entre los conceptos de un dominio. Si la relación es entre dos conceptos, se denomina relación binaria. Una relación binaria importante es Subclase de (Subclass-Of), que se usa para construir la taxonomía de clases.
- **Instancias, individuos o elementos**. Se usan para representar los elementos o individuos en una ontología. Las relaciones pueden ser también instancias.
- **Constantes**. Son valores que no sufren cambios en el tiempo.
- **Atributos o propiedades**. Describen las propiedades de las instancias y de los conceptos. Se distinguen dos tipos de atributos:
  1. **Atributos de instancias**. Describen los conceptos de las instancias, desde donde toman sus valores. Estos atributos están definidos en un concepto y son heredados por sus subconceptos e instancias.
  2. **Atributos de clases.** Describen los conceptos y toman los valores del concepto donde ellos están definidos. Estos atributos no son heredados ni por las subclases ni por las entidades.
- **Axiomas formales**. Son expresiones lógicas que siempre son realidad y normalmente son usadas para especificar las obligaciones en la ontología.
- **Reglas**. Son generalmente usadas para inferir conocimiento en una ontología; como los valores de los atributos, las relaciones entre las instancias, etc.

Se pueden establecer distintos tipos de ontologías atendiendo a diversos aspectos. Podemos destacar las siguientes clasificaciones, aunque existen otras muchas.

En la figura 3.7 se puede observar un ejemplo de ontología representada en UML. En ella aparecen diversos conceptos, como por ejemplo"árbol", "árbol forestal" y "árbol frutal", para los que se indican sus propiedades y se especifican las relaciones de agregación y asociación que tienen lugar entre ellos.

Tipos de Ontologías según su Expresividad

Lassila y McGuinness (2001) hablan del espectro de las ontologías como el espacio en el cual se presentan diferentes formas de representar conocimiento. Se pueden clasificar según el nivel de expresividad que usen en los siguientes tipos:

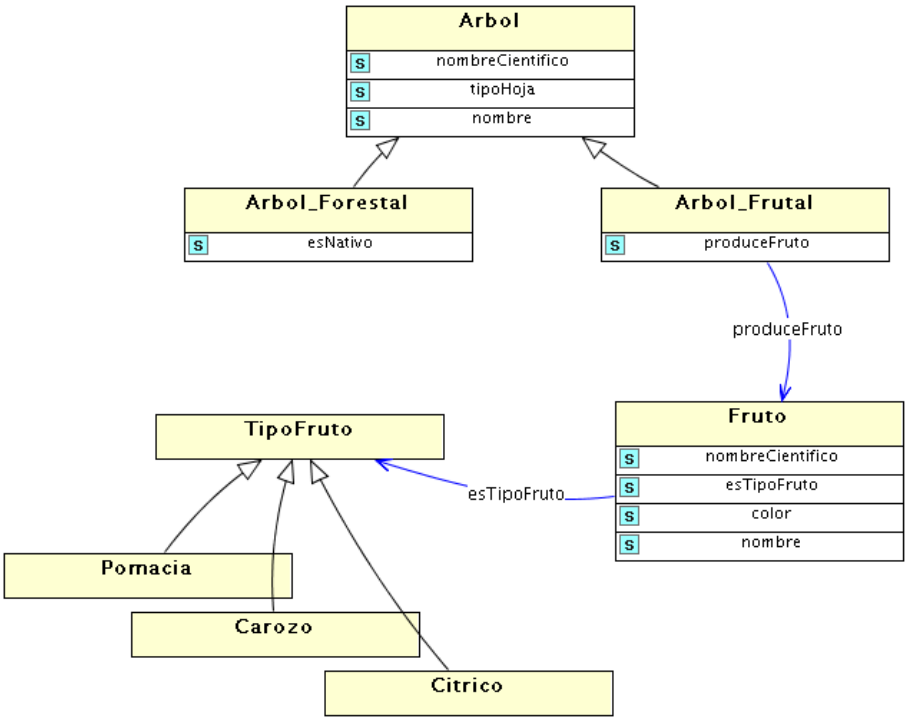


Figura 3.7: Ejemplo de ontología. Imagen extraída de `http://www.dcc.uchile.cl/~ekrsulov/slides/titulo/slide3-0.html`.

- **Vocabularios controlados:** Lista de términos.
- **Tesauros:** También se proporcionan relaciones entre términos.
- **Taxonomía informal:** Se define una jerarquía explícita (generalización y especialización), pero no estrictamente herencia.
- **Taxonomía formal:** Jerarquía explícita con herencia estricta.
- **Marcos:** Se definen clases compuestas por conjuntos de propiedades, las cuales se heredan por especialización.
- **Restricciones de valor:** Se restringen los valores de las propiedades.
- **Restricciones lógicas generales:** Los valores de las propiedades se restringen por medio de fórmulas lógicas o matemáticas aplicadas a valores de otras propiedades.
- **Restricciones de la lógica de primer orden:** Se permiten restricciones de la lógica de primer orden, y, por tanto, las relaciones son más detalladas.

En la figura 3.8 que aparece a continuación aparecen ubicadas los distintos tipos de ontologías de acuerdo con riqueza semántica (expresividad). Las ontologías situadas más hacia la izquierda disponen de semántica débil, mientras que las situadas a la derecha disponen de semántica fuerte. Aquellas situadas más hacia la izquierda representan una semántica simplificada, mientras que las situadas a la derecha disponen de semántica más compleja.

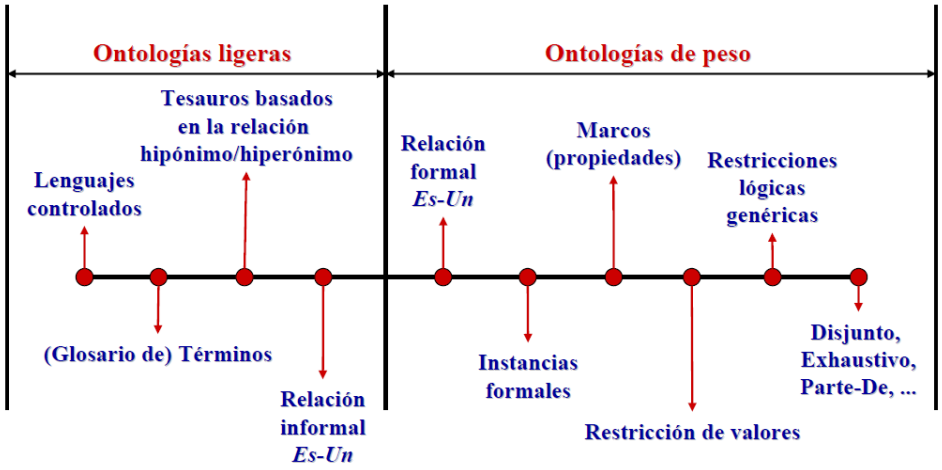


Figura 3.8: Clasificación de las ontologías acorde con su expresividad.

Tipos de Ontologías según su Nivel de Abstracción

Guarino (1997) clasifica las ontologías de acuerdo con su nivel de abstracción y con su relación con una tarea específica, diferenciando los siguientes tipos de ontologías:

- **Ontologías de alto nivel**: Describen conceptos muy generales como espacio, tiempo, evento, que son independientes de un problema o dominio particular.
- **Ontologías de dominio**: Describen el vocabulario relacionado con un dominio genérico mediante la especialización de conceptos introducidos en las ontologías de alto nivel (p.e. electrónica, mecánica, medicina).
- **Ontologías de tareas**: Describen el vocabulario relacionado con una tarea o actividad genérica mediante la especialización de las ontologías de alto nivel (p.e. "hipótesis" pertenece a la ontología de la realización de diagnósticos).

- **Ontologías de aplicación**: Son las más específicas, en ellas los conceptos corresponden a papeles jugados por entidades de dominio mientras realizan una determinada actividad. Su función más importante es la describir y delimitar un dominio de conocimiento, estableciendo un vocabulario en el que se describen de manera formal sus términos, propiedades de los mismos, así como las relaciones entre ellos.

3.3.7. Lenguajes Ontológicos

Aunque, en principio, la representación del conocimiento que se realiza con una ontología es independiente del lenguaje utilizado, se suele utilizar un enfoque muy próximo a la Lógica de Descripciones y, por extensión, a las representaciones basadas en estructuras de red. De hecho, en la actualidad el concepto de ontología es prácticamente equivalente al de representación en Lógica de Descripciones.

Los lenguajes ontológicos son vehículos para expresar ontologías de forma comprensible por las máquinas. Algunos de ellos han emergido en los últimos años en paralelo a la idea de Web Semántica, de forma que están orientados para tal tecnología. Históricamente han sido propuestos diferentes lenguajes por los desarrolladores de herramientas ontológicas, como son Ontolingua, KIF, LOOM, OKBC, OCML y FLogic.

En Gómez-Pérez et al. (2004) se puede encontrar un estudio de los lenguajes tradicionales citados en el párrafo anterior, así como lenguajes con marcas entre los que se pueden encontrar SHOE, XOL, RDFS, OIL, DAM + OIL y OWL. El lenguaje para definir e instanciar ontologías Web recomendado por la W3C es OWL.

3.3.7.1. OWL

En la actualidad, el lenguaje más extendido es OWL (McGuinness y van Harmelen, 2004). El objetivo que persigue OWL es proporcionar una sintaxis y una semántica (Patel-Schneider et al., 2004) para representar la información presente en documentos web de forma que pueda procesarse automáticamente. OWL recoge influencias de varios formalismos (Horrocks et al., 2003), entre los que destaca la Lógica de Descripciones. Al igual que en ésta, una base de conocimiento en OWL consiste en un conjunto de descripciones de clases, roles e instancias. OWL ha sido diseñado como lenguaje de ontologías Web para cubrir esta necesidad ya que, como se muestra a continuación, los demás lenguajes existentes cuentan con desventajas que no les permiten alcanzar la expresividad de la que dispone OWL.

- **XML** proporciona una sintaxis para la creación de documentos estructurados, pero no dispone de restricciones semánticas que operen en el contenido de estos documentos.
- **XML Schema** es un lenguaje que se utiliza para restringir la estructura de los documentos XML, además es capaz de definir los tipos de datos con los que se va a trabajar.
- **RDF** es un modelo de datos para describir recursos y relaciones entre ellos mediante el uso de tripletas (Entidad, Propiedad, Valor). Proporciona una semántica simple para ello y puede ser representado mediante una sintaxis XML. Carece de la formalidad de OWL.
- **RDF Schema** es utilizado para describir propiedades y clases de recursos RDF, a modo de vocabulario. Proporciona una semántica para la generalización y jerarquización tanto de propiedades como de clases, facilitando la creación de vocabularios y taxonomías. OWL es compatible con RDFS, de manera que cualquier ontología en RDFS es una ontología en OWL Full (descrito a continuación).
- **OWL** añade más vocabulario para describir propiedades y clases: Entre otros, relaciones entre clases, cardinalidad, igualdad, nuevos tipos de propiedades, características de propiedades, y clases enumeradas.

OWL ofrece tres niveles de expresividad que han sido definidos según las propiedades de la Lógica de Descripciones para asegurarse de que la implicación lógica es decidible. A grandes rasgos, la relación entre los tres niveles de OWL es la siguiente:

- **OWL Lite** es el nivel básico y está orientado a la creación de jerarquías de clasificación con restricciones simples. Por ejemplo, OWL Lite sólo permite restricciones de cardinalidad con valores 0 ó 1. Este sublenguaje es el menos expresivo, por lo que suele utilizarse para traducir a OWL tesauros y otras taxonomías. La ventaja principal de OWL Lite es que al ser más limitado, el razonamiento es más eficiente y resulta más sencillo trabajar con él.
- **OWL DL** es un sublenguaje más expresivo que el anterior, al tiempo que completo (todas las conclusiones son computables) y decidible (todos los cálculos terminan en tiempo finito). OWL DL incluye todos los constructores del lenguaje OWL pero sólo permite utilizarlos bajo ciertas restricciones. Por ejemplo, algunas de las restricciones más importantes que se introducen en OWL DL son:
  - Las clases no pueden considerarse individuos (y, en consecuencia, una clase no puede ser instancia de otra).

- La intersección entre el conjunto de los objetos y los tipos de datos concretos debe ser vacía (por lo tanto, no pueden definirse nuevos tipos de datos).
- Ciertos calificativos (inversa, inversa funcional, simétrica, transitiva) no pueden aplicarse sobre las propiedades de tipos de datos.

- **OWL Full** ofrece la mayor expresividad, aunque a costa de convertirse en un lenguaje no decidible. OWL Full no proporciona constructores adicionales a OWL DL, pero permite utilizar todos los de éste sin restricciones. De hecho, OWL Full no es un sublenguaje de OWL, pues engloba a todos los anteriores.

Las clases en OWL agrupan a individuos que comparten propiedades. Una descripción de clase asigna un nombre a la clase (URI) y establece qué individuos forman parte de ella. Las descripciones se pueden combinar para componer axiomas complejos utilizando diversos constructores dependiendo del nivel de expresividad que se pretenda para la ontología.

Existen diversas sintaxis para OWL: RDF/XML, OWL/XML, y sintaxis funcional OWL; siendo la primera la más extendida. Un documento OWL expresado con la sintaxis RDF/XML consiste en un conjunto de marcas XML con la sintaxis y la semántica ordenadas por el estándar. Habitualmente, un fichero OWL consta de tres partes: Una cabecera, un conjunto de definiciones de clases y propiedades (junto con sus respectivas restricciones) y una serie de aserciones sobre individuos.

En la cabecera se especifican algunas consideraciones previas acerca de la ontología.

```
<rdf:RDF xmlns="http://www.ontologies.org/pizza/pizza.owl#"
     xml:base="http://www.ontologies.org/pizza/pizza.owl"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#">
    <owl:Ontology rdf:about="">
        <owl:versionInfo xml:lang="en">
        . . .
     </owl:Ontology>
</rdf:RDF>
```

Ejemplo 3.9: Definición de Espacios de Nombres en una ontología en OWL.

En el ejemplo 3.9 se puede observar cómo se incluyen las definiciones de los espacios de nombres que se utilizarán en el documento o los modelos adicionales que se importan desde el actual.

Los operadores para la definición de clases en OWL son los que se señalan a continuación, teniendo en cuenta que sólo se consideran aquellos que pueden utilizarse en los niveles DL y Full:

- Herencia: *rdfs:subClassOf.*
- Equivalencia: *owl:equivalentClass.*
- Disyunción: *owl:disjointWith.*
- Enumeración de individuos: *owl:oneOf (DL), owl:DataRange (DL).*
- Restricciones de valor sobre propiedades: o*wl:someValuesFrom, owl:allValuesFrom, owl:hasValue (DL).*
- Restricciones de cardinalidad sobre propiedades: owl:cardinality, owl:minCardinality, owl:maxCardinality.
- Operaciones de conjuntos: *owl:unionOf (DL), owl:intersectionOf, owl:complementOf (DL).*

```
<owl:Class rdf:about="#WhiteLoire">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#madeFromGrape" />
      <owl:allValuesFrom>
        <owl:Class>
          <owl:oneOf rdf:parseType="Collection">
            <owl:Thing rdf:about="#CheninBlancGrape" />
            <owl:Thing rdf:about="#PinotBlancGrape" />
            <owl:Thing rdf:about="#SauvignonBlancGrape" />
          </owl:oneOf>
        </owl:Class>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
      <owl:Class rdf:ID="#Wine"/>
  </rdfs:subClassOf>
</owl:Class>
```

Ejemplo 3.10: Definición de la clase WhiteLoire.

El ejemplo 3.10 anterior muestra cómo se definiría en OWL una clase llamada "WhiteLorie", descendiente de "Wine", con una restricción sobre la propiedad "madefromGrape".

Las propiedades de OWL se utilizan para definir relaciones entre individuos de la ontología (propiedades de objetos: *owl:ObjectProperty*) o individuos y valores (propiedades de tipos de datos: *owl:DatatypeProperty*).

Los *axiomas de propiedades*, que describen sus características, se construyen utilizando los siguientes operadores:

- Herencia: *rdfs:subPropertyOf.*
- Rango y dominio: *rdfs:range, rdfs:domain.*
- Relaciones con otras propiedades: *inversa (owl:inverseOf), equivalencia (owl:equivalentProperty).*
- Limitaciones de cardinalidad global: *funcional (owl:FunctionalProperty), funcional inversa (owl:InverseFunctionalProperty).*
- Limitaciones lógicas: *simetría (owl:SymmetricProperty), transitividad (owl:TransitiveProperty).*

```
<owl:ObjectProperty rdf:ID="locatedIn">
   <rdf:type rdf:resource="&owl;TransitiveProperty" />
      <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl
         #Thing" />
      <rdfs:range rdf:resource="#Region" />
</owl:ObjectProperty>
```

Ejemplo 3.11: Definición de una propiedad en OWL.

En el ejemplo 3.11 anterior, se puede observar cómo se realiza la definición de la propiedad "locatedIn".

En el caso de los individuos, los axiomas suelen llamarse hechos y en ellos se establece un nombre para el individuo (si no es anónimo), información sobre la clase a la que pertenece y valores para sus propiedades. OWL no asume que dos individuos con nombres diferentes sean distintos -*suposición conocida como UNA (*Unique Name Assumption*)*- por lo que en los hechos puede incluirse información sobre la identidad del individuo que se está describiendo (owl:sameAs, owl:differentFrom, owl:AllDifferent).

```
<WhiteLoire rdf:ID="Wine0001">
     <owl:sameAs rdf:resource="Vino0007"/>
</WhiteLoire>
```

Ejemplo 3.12: Definición de una individuo o instancia en OWL.

En el ejemplo 3.12 anterior, se puede observar cómo se realiza la definición de un individuo perteneciente a la clase "WhiteLoire".

Para facilitar la legibilidad de la ontología, aparte de las descripciones de clases, relaciones e individuos, un documento OWL puede incluir un número cualquiera de anotaciones, así como información adicional en la cabecera sobre autoría, número de versión, etc.

3.3.7.2. OWL 2

OWL 2 (Motik et al., 2008c) es una nueva especificación que refina y extiende OWL. Es una recomendación de la W3C propuesta en Octubre de 2009 (Hitzler et al., 2009).

OWL 2 tiene una estructura muy similar a la de OWL, que se ha pasado a llamar OWL 1. El papel central que tiene la sintaxis RDF/XML (Beckett, 2004a) no ha cambiado, y también mantiene otras sintaxis como OWL/XML o la sintaxis funcional. Las relaciones entre las semánticas "Directa" (Motik et al., 2008b) y la basada en RDF (Schneider, 2009) tampoco han cambiado. Y lo que es lo más importante, la compatibilidad con OWL 1 es completa a todos los efectos: Todas las Ontologías OWL 1 continúan siendo Ontologías válidas en OWL 2, con idénticas inferencias en la práctica.

OWL 2 añade nueva funcionalidad con respecto a OWL 1. Algunas de sus nuevas características son sintácticas mientras que otras ofrecen nueva expresividad (Golbreich y Wallace, 2008), entre las que se incluyen:

- Llaves.
- Cadenas de propiedades.
- Nuevos tipos de datos y rangos.
- Restricciones cualificadas de cardinalidad.
- Propiedades asimétrica, reflexiva y disjunta.
- Capacidad de anotación mejorada.

OWL 2 también ofrece tres nuevos sublenguajes denominados perfiles (Motik et al., 2008a) y una nueva sintaxis llamada Manchester (Horridge y Patel-Schneider, 2009). Además, se han relajado algunas restricciones aplicables a OWL DL; como resultado, el conjunto de grafos RDF que pueden ser manejados por los razonadores de Lógica Descriptiva (Baader et al., 2003) es ligeramente mayor en OWL 2.

- **El perfil OWL 2 EL** es particularmente útil en aplicaciones que emplean ontologías con gran cantidad de propiedades y/o clases. Este perfil dispone del poder expresivo que necesitan muchas ontologías junto con el subconjunto de OWL 2 para el cual los problemas de razonamiento básico pueden realizarse en tiempo de cómputo polinomial con respecto al tamaño de la ontología. Este perfil también dispone de algoritmos de razonamiento que permiten una implementación altamente escalable. El acrónimo *EL* denota que el perfil tiene como base la familia de lógica descriptiva (EL++), la cual proporciona solamente cuantificación existencial.

- **El perfil OWL 2 QL** está dirigido a aplicaciones que utilizan grandes volúmenes de instancias de datos, donde el tiempo de respuesta a las consultas es la tarea de razonamiento más importante. Las respuestas a las consultas se pueden implementar usando sistemas de bases de datos relacionales convencionales. En OWL 2 QL, se pueden usar algoritmos de tiempo de cómputo polinomial para implementar los problemas de consistencia de ontologías y de subsunción de expresiones de clases. Su poder expresivo es bastante limitado, aunque incluye la mayoría de las características principales de modelos conceptuales tales como los diagramas de clases UML y Entidad-Relación. El acrónimo *QL* denota el hecho de que la respuesta a consultas puede ser realizada mediante la reescritura de consultas en un lenguaje de consultas estándar (Query Language).

- **El perfil OWL 2 RL** está dirigido a aplicaciones que requieren un razonamiento escalable sin sacrificar demasiado su potencia expresiva. Está diseñado para dar cabida a aplicaciones OWL 2 capaces de utilizar la expresividad completa del lenguaje consiguiendo eficiencia, así como aplicaciones RDFS que necesitan expresividad añadida. Los sistemas de razonamiento OWL 2 RL se puede implementar utilizando motores de razonamiento basados en reglas. La consistencia de una ontología, la satisfacibilidad de una expresión de clase, la subsunción de una expresión de clase, la comprobación de instancias y los problemas de respuesta a consultas conjuntivas pueden ser resueltos en el tiempo de cómputo polinomial con respecto al tamaño de la ontología. Las siglas RL refleja el hecho de que el razonamiento en este perfil se puede implementar utilizando un lenguaje de reglas estándar (Rule Language).

3.3.8. SPARQL

SPARQL es un acrónimo recursivo del inglés **S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage. Se trata de un lenguaje estandarizado para la con-

sulta de grafos RDF, normalizado por el RDF Data Access Working Group (DAWG) del W3C. Es una tecnología clave en el desarrollo de la Web Semántica que se constituyó como Recomendación oficial del W3C el 15 de Enero de 2008 (Prud'hommeaux y Seaborne, 2008).

Al igual que sucede con SQL, es necesario distinguir entre el lenguaje de consulta y el motor para el almacenamiento y recuperación de los datos. Por este motivo, existen múltiples implementaciones de SPARQL ligados a diferentes entornos de desarrollo y plataformas tecnológicas.

En un principio SPARQL únicamente incorpora funciones para la recuperación sentencias RDF. Sin embargo, algunas propuestas también incluyen operaciones para el mantenimiento (creación, modificación y borrado) de datos. Algunas de sus características nos permiten:

- Extraer información de URIs, literales y nodos vacíos.
- Obtener subgrafos RDF.
- Construir nuevos grafos RDF basados que información de grafos que devuelve una query.

Las consultas más sencillas que podemos construir necesitan de las siguientes cláusulas:

- **Cláusula "SELECT"**: Identifica las variables que se requiere que aparezcan en el resultado de la consulta.
- **Cláusula "WHERE"**: Está formada por tripletas que establecen la condición de búsqueda.

A continuación, el ejemplo 3.13 muestra la consulta que necesitamos realizar para saber el título de un libro.

```
1 PREFIX  dc: <http://purl.org/dc/elements/1.1/>
2 SELECT  ?title
3 WHERE   { <http://ejemplo.org/libros> dc:title ?title }
```

Ejemplo 3.13: Consulta en SPARQL.

3.3.9. RIF

El grupo de trabajo RIF (Kifer y Boley, 2010) fue constituido en 2005 por el consorcio W3C para crear un estándar para el intercambio de reglas entre sistemas, en particular entre motores de reglas web. Los trabajos iniciales

del grupo se centraron en el intercambio en lugar de intentar desarrollar un lenguaje de reglas unificado. Incluso esta tarea era inabordable, por lo que la solución que adoptó el grupo de trabajo RIF fue el diseño de una familia de lenguajes, llamados dialectos, con una especificación rigurosa en su sintaxis y en su semántica. Un dialecto RIF es un lenguaje basado en reglas con sintaxis XML y una semántica bien definida.

La familia de dialectos RIF pretende ser *uniforme* y *extensible*. Uniformidad RIF significa que se espera que los dialectos RIF compartan, tanto como sea posible, los aparatos sintácticos y semánticos existentes. Extensibilidad significa que debería ser posible para expertos la definición de un nuevo dialecto RIF como extensión sintáctica de un dialecto existente, con nuevos elementos correspondientes a la funcionalidad adicional deseada. Estos nuevos dialectos RIF serían no estándar en el momento de su definición, pero podrían eventualmente llegar a convertirse en estándar.

RIF ofrece más que sólo un formato, aunque el concepto de formato es esencial para la forma en la que se utiliza RIF. El medio de intercambio entre los sistemas de reglas diferentes es XML. La idea central para el intercambio de reglas a través de RIF es que los diferentes sistemas proporcionarán mapeos sintácticos para sus lenguajes nativos y viceversa. Se requiere que estos mapeos mantengan la semántica, y los conjuntos de reglas puedan ser comunicados desde un sistema a otro siempre y cuando los sistemas puedan hablar a través de un dialecto apropiado, que sea compatible con ambos.

El grupo de trabajo RIF se ha centrado en dos tipos de dialectos: Dialectos basados en la lógica y dialectos basados en reglas con acciones. Generalmente, los dialectos basados en la lógica incluyen lenguajes que utilizan algún tipo de lógica, ya sea de primer orden o los que utilizan lenguajes de programación lógica. Los dialectos de reglas con acciones incluyen sistema de producción de reglas, tales como Jess [2], Drools [3] y JRules [4], así como reglas reactivas, tales como RuleML [5] y XChange (Bry y lavinia P Atranjan, 2005). Se necesitaron cuatro años para desarrollar los tres dialectos que se han convertido en recomendados por las W3C, y que se explican a continuación:

- RIF-BLD (*RIF Basic Logic Dialect*, Dialecto RIF de Lógica Básica) (Boley y Kifer, 2010a) es uno de los dos dialectos principales, y es el dialecto principal basado en lógica desarrollado por el grupo. Técnicamente, este dialecto corresponde a la lógica de Horn con varias exten-

[2] `http://www.jessrules.com`

[3] `http://www.jboss.org/drools`

[4] `http://www-01.ibm.com/software/integration/business-rule-management/jrules`

[5] `http://ruleml.org`

siones sintácticas (estructura de la sintaxis y argumentos de predicados con nombre) y semánticas (tipos de datos y predicados definidos externamente). Este dialecto cubre muchos de los sistemas de reglas existentes a pesar de que puede no ser lo suficientemente expresivo para algunos sistemas de reglas.

- RIF-PRD (*RIF Production Rule Dialect*, Dialecto RIF de Producción de Reglas) (de Sainte Marie et al., 2010) es el otro dialecto principal del grupo, encargado de capturar los aspectos principales de varios sistemas de producción de reglas. Existe interés en la tecnología de producción de reglas por parte de la industria. La producción de reglas, como se está practicando actualmente en sistemas como Jess o JRules, se define usando mecanismos computacionales Ad-Hoc, los cuales no se basan en la lógica.

- RIF Core (Harold Boley y Reynolds, 2010) es el dialecto central, es un subconjunto de RIF-BLD y RIF-PRD basado en RIF-DTB (*RIF Datatypes and Built-ins*, Tipos de Dato y Empotrados RIF) que permite el intercambio de reglas entre dialectos de reglas lógicas y dialectos de reglas de producción. RIF-Core corresponde a una lógica Horn carente de símbolos de función con extensiones que posibilitan su compatibilidad con objetos y estructuras in F-logic, IRIs para conceptos, y tipos de datos para Esquemas XML.

```
1  Document(
2    Prefix(cpt <http://example.com/concepts#>)
3    Prefix(ppl <http://example.com/people#>)
4    Prefix(bks <http://example.com/books#>)
5
6    Group
7    (
8      Forall ?Buyer ?Item ?Seller (
9          cpt:buy(?Buyer ?Item ?Seller) :- cpt:sell(?Seller ?
               Item ?Buyer)
10     )
11
12     cpt:sell(ppl:John bks:LeRif ppl:Mary)
13   )
14 )
```

Ejemplo 3.14: Sintaxis de Presentación en RIF-Core.

En el ejemplo 3.14 se puede observar que el hecho de que "Mary compra el libro LeRif a John" puede ser lógicamente derivado usando *Modus Ponens* a partir del hecho de que "John vende el libro LeRif a Mary". Para su representación se está usando la sintaxis de presentación RIF-Core.

Además de dialectos anteriores, el Grupo de Trabajo RIF ha desarrollado los siguientes documentos que también son recomendaciones de la W3C:

- RIF-FLD (*RIF Framework for Logic Dialects*, Marco de trabajo RIF para Dialectos Lógicos) (Boley y Kifer, 2010b) no es un dialecto en sí mismo, sino un marco de extensibilidad lógica general. Fue introducido con el fin de reducir drásticamente la cantidad de esfuerzo necesario para definir y comprobar nuevos dialectos lógicos que amplíen las capacidades de RIF-BLD.
- En el documento de DeBruijn (2010) se describe cómo conseguir la interoperabilidad entre RIF y otros estándares de la Web Semántica. En él se define la sintaxis y la semántica de los lenguajes combinados RIF+RDF y RIF+OWL2.
- En el documento de Polleres et al. (2010) acerca de RIF-DTB, se identifican los tipos de dato más comunes, las funciones integradas y los predicados, y se define su semántica de forma precisa con objeto de preservar la semántica en el intercambio de reglas.

3.4. Herramientas de la Web Semántica

En los últimos años, han aparecido un gran número de herramientas dentro del ámbito de la Web Semántica. Como es lógico abundan las herramientas para trabajar con ontologías ya que éstas juegan un papel esencial: Son el elemento clave para reutilizar y compartir conocimiento.

En particular abundan los entornos para la construcción y uso de ontologías. El uso de herramientas de ayuda es importante tanto para el proceso de desarrollo de ontologías (construcción, anotación, combinación) como para el uso de las ontologías en aplicaciones tales como comercio electrónico, gestión de conocimiento y la Web Semántica.

3.4.1. Tipos de Herramientas

En algunas webs se recopilan numerosas herramientas, como por ejemplo en `http://www.mkbergman.com/sweet-tools-simple-list`. En ésta aparecen alrededor de 1100 herramientas clasificadas en diversas categorías. Según Gómez-Pérez et al. (2002) se pueden organizar en las siguientes categorías:

- **Desarrollo de Ontologías:** En este grupo se incluyen herramientas, entornos y suites que se pueden usar para construir una nueva ontología desde cero o reusando otras ontologías. Además de la funcionalidad

de edición y navegación, estas herramientas suelen incluir documentación sobre las ontologías, importación y exportación desde diferentes formatos, visor gráfico de las ontologías construidas, librerías, motor de inferencia asociado, etc.

- **Evaluación de Ontologías:** Aparecen como herramientas de soporte que aseguran que tanto las ontologías como su tecnología asociada disponen de un determinado nivel de calidad. Asegurar la calidad es sumamente importante para evitar problemas en la integración de ontologías con aplicaciones industriales con tecnología basada en ontologías.
- **Combinación e Integración de Ontologías:** Estas herramientas han aparecido para resolver el problema de integrar diferentes ontologías en el mismo dominio. Esta necesidad aparece cuando dos organizaciones o compañías se fusionan, o cuando es necesario obtener una ontología de mejor calidad a partir de otras ontologías del mismo dominio.
- **Herramientas de Anotación Basadas en Ontologías:** Han sido diseñadas para permitir a los usuarios insertar y mantener de forma (semi)automática marcados basados en ontologías en páginas Web. La mayoría de estas herramientas están integradas en el entorno de desarrollo de ontologías.
- **Almacenamiento y Consulta de Ontologías:** Estas herramientas se han creado para permitir a los usuarios usar y consultar ontologías de forma fácil. Debido a la gran aceptación y uso de la Web como plataforma de comunicación de conocimiento, han aparecido en este contexto nuevos lenguajes de consulta de ontologías.
- **Aprendizaje de Ontologías Basado en Procesamiento de Lenguaje Natural (NLP):** Son herramientas que se usan para obtener (semi)automáticamente ontologías a partir de textos en lenguaje natural.

Se hace necesario completar la clasificación anterior con nuevas categorías acorde a las características de las herramientas que están apareciendo en el contexto de la Web Semántica. Así pues, proponemos:

- **Razonadores:** Herramientas que pueden realizar tareas de razonamiento, típicamente basado en RDF o OWL, o algún motor de inferencia. Algunas de estas herramientas forman parte de una herramienta más compleja (Jena, Spatial Oracle) mientras que otras se pueden añadir a los entornos existentes para añadir y/o mejorar la capacidad de razonamiento (Pellet).

- **Librerías de Programación o APIs:** Herramientas cuya principal objetivo es definir y/o implementar una API para manejar datos de la Web Semántica (RDF, OWL, etc).
- **Herramientas de Desarrollo Web Basado en Ontologías:** Se trata de herramientas que incluyen ontologías como modelo subyacente en el que se apoya el diseño y desarrollo de sitios y portales Web.
- **Herramientas de Desarrollo de Servicios Web Semánticos:** En los servicios web semánticos se usan ontologías para describir la funcionalidad y características del servicio web: Sus entradas y salidas, las condiciones necesarias para que se puedan ejecutar, los efectos que producen, o los pasos a seguir cuando se trata de un servicio compuesto.

De forma complementaria a las categorías simples mencionadas anteriormente existen conjuntos de herramientas o suites, que incluyen varios aspectos del tratamiento de ontologías, comúnmente conocidas como **Herramientas de Gestión de Ontologías** (Martin-Recuerda et al., 2004).

En el artículo titulado "*Una Panorámica Actual de Software para Trabajar con Ontologías*" (Navarro-Galindo y Samos, 2007b) se realizó un estudio detallado de las herramientas software que existían en aquel momento y que eran representativas dentro de las categorías que se han presentado anteriormente. Casi todas estas herramientas han perdurado hasta hoy día, manteniendo una constante evolución y adaptación a las tecnologías actuales.

En la tablas que aparecen a continuación en las figuras 3.9 y 3.10, se puede encontrar de forma resumida información sobre dichas herramientas. Destacar que, en el caso de tratarse de herramientas que desarrollan varias funcionalidades, se ha optado por clasificarlas como pertenecientes a la categoría donde resultan más relevantes.

HERRAMIENTA (APARTADO)	CATEGORÍA	OTRAS FUNCIONALIDADES	LENGUAJE DE ONTOLOGÍAS
Ontolingua (3.1)	Desarrollo	API de integración de ontologías con agentes software.	Ontolingua, IDL, KIF, IPS,LOOM, OKBC, PROLOG
OntoStudio (3.2)	Desarrollo	Representación gráfica. Integración con SGBD.	F-LOGIC, OXML, OWL, RDF(S)
Protégé (3.3)	Desarrollo	Permite programar plugins para ampliar su funcionalidad mediante API Java	XML, RDF(S),OWL, CLIPS, N-TRIPLE, N3, TURTLE
Swoop (3.4)	Desarrollo	Consultas RDQL usando PELLET Arquitectura plugin. Anotación	XML, RDF(S), OWL, TURTLE, Sintaxis abstracta
WebODE (3.5)	Desarrollo	Chequeo de consistencia de ontologías. API de acceso a las ontologías.	XML, RDF(S), DAML+OIL, CARIN, Flogic, Prolog, Jess
WebOnto (3.6)	Desarrollo	Permite trabajo colaborativo mediante mensajes y anotaciones.	OCML, Ontolingua, GXL, RDF(S), OIL
ONE-T (4.1)	Evaluación	Integrada en Ontolingua.	Ontolingua, IDL, KIF, IPS,LOOM, OKBC, PROLOG
OntoClean (4.2)	Evaluación	Integrada en WebODE.	XML, RDF(S), DAML+OIL, CARIN, Flogic, Prolog, Jess
Chimaera (5.1)	Comb. e Integ.	Independiente del editor.	OKBC
PROMPT (5.2)	Combinación e integración	Dispone de guía para combinación e integración de ontologías.	XML, RDF(S),OWL, CLIPS, N-TRIPLE, N3, TURTLE
ODEMerge (5.3)	Combinación e integración	Integrada en WebODE.	XML, RDF(S), DAML+OIL, CARIN, Flogic, Prolog, Jess
KIM (6.1)	Anotación	Población automática de ontologías.	OWL
OntoMat (6.2)	Anotación	Interfaz plugin para ampliación	OWL
Kowari (7.1)	Almacenamiento	SPARQL, Tucana Query Language	RDF, OWL
Sesame (7.2)	Almacenamiento y consulta	Independiente del SGBD	RDF, OWL
Owlim (7.3)	Almacenamiento y consulta	Integrada en KIM y también disponible de forma independiente.	OWL
KEA (8.1)	Aprendizaje	En base a vocabularios controlados	RDF(S)
TextToOnto (8.2)	Aprendizaje	Maneja texto libre, semiestructurado, ontologías, diccionarios y BD	DAM+OIL, RDF(S)
Jena (9.1)	Gestión de ontologías	Proporciona APIs de desarrollo para RDF Y OWL así como SPARQL	RDF, OWL, N3, N-Triples
Kaon2 (9.2)	Gestión de ontologías	Proporciona APIs, interfaz DIG, SPARQL y razonador propio	OWL-DL, SWRL, F-Logic
Corese (11.1)	Desarrollo Web	Proporciona API de desarrollo y lenguaje de consultas SPARQL.	RDF, RDF(S)
OntoWeaver (11.2)	Desarrollo Web	Creación de portales y sitios Web basados en ontologías	RDF, RDF(S)
OntoWebber (11.3)	Desarrollo Web	Creación de portales y sitios Web basados en ontologías	RDF, DAM+OIL
DIP Ontology Manag. Suite (12.2)	Servicios Web Semánticos	Propone arquitectura DIP. Integración de ontologías con Serv. Web Semánticos	WSML
DOME (12.1)	Servicios Web Semánticos	Poporciona API de acceso a repositorios y motores de inferencia	WSML
pOWL (9.3)	Gestión de ontologías	Dispone de Ontowiki, plataforma de desarrollo de Bases de Conocimiento	OWL
WSMO Studio (12.3)	Servicios Web Semánticos	Entorno integrado WSMO	WSML, OWL-DL, RDF
WonderWeb OWLAPI (10)	API	Parseo, inferencia, análisis de tipos	OWL
SchemaWeb (13.1)	Buscadores	Acceso y registro de esquemas ontolog.	OWL, RDF, DAM+OIL
Swoogle (13.1)	Buscadores	Indexación, recuperación y organiz.	OWL, RDF, DAM+OIL
Proton (13.2)	Meta-ontologías	-	OWL
Sumo (13.2)	Meta-ontologías	-	OWL, LOOM, Protégé
FaCT++ (13.3)	Razonadores	Integración DIG con otras herramientas	OWL
Pellet (13.3)	Razonadores	Integración DIG con otras herramientas	OWL
RacerPro (13.3)	Razonadores	Integración DIG con otras herramientas	OWL, RDF, SWRL

Figura 3.9: Comparativa de herramientas de la Web Semántica I.

HERRAMIENTA (APARTADO)	PLATAF.	LENG. DES.	DESARROLLADORES	VERS.	FECHA	LICENCIA
Ontolingua (3.1)	Acceso Web	?	KSL (Stanford University)	1.0.650	14/10/2002	Acceso gratuito
OntoStudio (3.2)	Aplicación Win32	?	Ontoprise	1.6	22/12/2006	Software propietario
Protégé (3.3)	Independiente	Java	SMI (Stanford University)	3.2.1	2007	Freeware
Swoop (3.4)	Independiente	Java	MIND lab at University of Maryland	2.2.2	18/01/2007	GNU/GPL
WebODE (3.5)	Acceso Web	Java CORBA RMI Minerva	Ontologycal Group de la UPM	2.0.9	01/11/2003	Acceso gratuito y bajo licencia
WebOnto (3.6)	Applet Java	Java	KMI (Open University)	2.3	05/2001	Acceso gratuito
ONE-T (4.1)	Ontolingua	?	KSL (Stanford University)	-	-	Acceso gratuito
OntoClean (4.2)	WebODE	Java	UPM	-	-	Acceso gratuito y bajo licencia
Chimaera (5.1)	Ontolingua	?	KSL (Stanford University)	-	-	Acceso gratuito
PROMPT (5.2)	Plugin para Protégé	Java	Stanford University	2.4.8	20/06/2005	Freeware
ODEMerge (5.3)	WebODE	Java	UPM	-	-	Acceso gratuito y bajo licencia
KIM (6.1)	Apache Tomcat	Java	Ontotext Lab	1.7.12. 15	15/12/2006	S. Propietario
OntoMat (6.2)	Java Web Start	Java	Ontoagent Project	0.8	5/10/2004	Freeware
Kowari (7.1)	Independiente	Java	Sourceforge.net	1.1	09/01/2006	Open Source
Sesame (7.2)	Independiente	Java	Openrdf.org	2.0beta 3	6/04/2007	GNU/GPL
Owlim (7.3)	Independiente	Java	Ontotext Lab	2.9rc1	04/04/2007	Open Source
KEA (8.1)	Independiente	Java	The University of Waikato	4.1	07/11/2006	GNU/GPL
TextToOnto (8.2)	Independiente	Java	Instituto AIFB en la Universidad de Karlsruhe	1.0	09/11/2004	GNU/GPL
Jena (9.1)	Independiente	Java	HP Labs	2.5.2	18/01/2007	GNU/GPL
Kaon2 (9.2)	Independiente	Java	IPE, FZI, AIFB, IMG y U. de Manchester	2	27/02/2007	Gratis para uso no comercial
Corese (11.1)	Independiente	Java	INRIA	2.2.2	12/12/2006	Freeware
OntoWeaver (11.2)	Apache Tomcat + aplicación	Java	KMI Institute	Beta	15/03/2005	Open Source
OntoWebber (11.3)	Apache Tomcat + aplicación	Java	Universidad de Stanford	1.0	14/10/2002	Open Source
DIP Ontology Manag. Suite (12.2)	Independiente	Java	Project DIP	Protot.	30/06/2006	Open Source
DOME (12.1)	Independiente	Java	Ontology Management Working Group	0.2.0	18/12/2005	MIT License
pOWL (9.3)	Apache, Mysql, PHP	PHP	Sourceforge.net	0.94	21/03/2007	GNU/GPL
WSMO Studio (12.3)	Independiente	Java	Proyectos EU IST, DIP, SemanticGov y SUPER	0.5.5	20/03/2007	GNU/GPL
WonderWeb OWLAPI (10)	Independiente	Java	Proyecto WonderWeb	1.4.3	12/04/2006	GNU/GPL
SchemaWeb (13.1)	-	-	Vicsoft	-	2004	Acceso gratuito
Swoogle (13.1)	-	-	Ebiquity Group	-	2006	Acceso gratuito
Proton (13.2)	Independiente	OWL	SEKT Project	-	04/2005	Freeware
Sumo (13.2)	Independiente	KIF	IEEE Ontology Group	-	2003	Freeware
FaCT++ (13.3)	Independiente	C++	Universidad de Manchester	1.5	07/12/2006	Open Source
Pellet (13.3)	Independiente	Java	Mindswap lab	1.4	16/03/2007	Open Source
RacerPro (13.3)	Windows/ UNIX/Mac	?	Racer Systems GmbH	1.9	2007	Propietario

Figura 3.10: Comparativa de herramientas de la Web Semántica II.

3.4.2. Herramientas Usadas en la Tesis

Este apartado se dedica al estudio de las herramientas que han sido utilizadas durante las implementaciones que han conducido al desarrollo de la presente tesis. Al respecto, cabe puntualizar que, en particular, las herramientas de anotación semántica no aparecen aquí ya que se estudiarán más adelante en un apartado específico, el 4.4, perteneciente a la sección 4 que trata sobre todo lo relacionado con la anotación semántica de documentos Web.

3.4.2.1. Protégé

Se trata de una herramienta muy difundida y bien considerada por los desarrolladores del ámbito de ingeniería de ontologías. Así lo confirma el estudio realizado por Khondoker y Müller (2010), el cual concluye con que Protégé es la herramienta más dominante e independiente del dominio, con un uso estimado de un 75 % del total de usuarios encuestados. Una de las razones por la que un número elevado de desarrolladores tiende a usar Protégé es que dispone de ayuda en línea, mediante lista de correo. Más del 50 % de los encuestados piensan que se siente bien desarrollando con Protégé, piensan que el desarrollo de ontologías usando dicha herramienta es interesante y que la ayuda que ofrece es adecuada.

Protégé[6] es una plataforma gratuita de código abierto, que proporciona una herramienta con la que una comunidad creciente de usuarios puede construir modelos de dominio y aplicaciones basadas en conocimiento con ontologías.

En su núcleo, Protégé implementa un rico conjunto de estructuras de modelado de conocimiento y actividades que ayudan a la creación, visualización y manipulación de ontologías en varios formatos de representación. Protégé puede ser personalizado para proporcionar ayuda en dominios para la creación de modelos de conocimiento y entrada de datos.

Además, la funcionalidad que ofrece Protégé puede ser ampliada mediante extensiones, por medio de la arquitectura plugin y de una API Java para construir herramientas y aplicaciones basadas en el conocimiento.

La herramienta Protégé se puede usar para el desarrollo de ontologías que describen conceptos y relaciones importantes en un dominio particular, proporcionando un vocabulario de ese dominio, así como una especificación computerizada del significado de los términos utilizados en el vocabulario. Las ontologías se extienden desde las taxonomías y clasificaciones, a los es-

[6] http://protege.stanford.edu/

quemas de bases de datos, y a las teorías totalmente axiomáticas. En los últimos años, las ontologías se han adoptado en la comunidad científica y empresarial como una forma de compartir, reutilizar y procesar conocimiento de dominio. Las ontologías ahora son fundamentales para muchas aplicaciones como por ejemplo portales de conocimiento científico, gestión de información e integración de sistemas, comercio electrónico y servicios web semánticos.

La plataforma Protégé soporta dos formas principales de modelar ontologías:

- **El Editor de Marcos:** Proporciona una interfaz de usuario completa y un servidor de conocimiento para ayudar al usuario en la construcción y almacenado de ontologías basadas en frames. También ofrece la posibilidad de personalizar los formularios de entrada de datos y la introducción de datos de instancias. Implementa un modelo de conocimiento compatible con el protocolo OKBC (*Open Knowledge Base Connectivity*, Conectividad Abierta de la Base de Conocimiento)[7]. Dispone de las siguientes características:
  - Incluye un amplio conjunto de elementos de interfaz de usuario que pueden ser personalizados para facilitar a los usuarios la entrada de datos en los formularios del modelo de conocimiento.
  - Tiene una arquitectura extensible mediante plugins que puede incorporar elementos personalizados, como pueden ser componentes gráficos (gráficos y tablas), multimedia (sonidos, imágenes y vídeo), diversos formatos de almacenamiento (RDF, XML, HTML) y soporte para herramientas adicionales (gestión de ontologías, visualización de ontologías, inferencia y razonamiento, etc).
  - Una API Java que hace posible que los plugin y otras aplicaciones puedan acceder, usar y mostrar ontologías creadas con el editor de marcos.
- **El Editor de OWL:** Es una extensión de Protégé que soporta el lenguaje OWL. El editor permite a los usuarios realizar las tareas que se indican a continuación:
  - Cargar y salvar ontologías OWL y RDF.
  - Editar y visualizar clases, propiedades y reglas SWRL.
  - Definir características lógicas de las clases con expresiones OWL.
  - Ejecutar razonadores, así como clasificadores de Lógica Descriptiva.

[7] `http://www.ai.sri.com/~okbc`

- Editar individuos OWL para la Web Semántica.

En cuanto al aspecto que presenta la herramienta y la funcionalidad que ofrece, se puede destacar lo siguiente:

- La apariencia que ofrece el Editor de Marcos es la que aparece en la figura 3.11

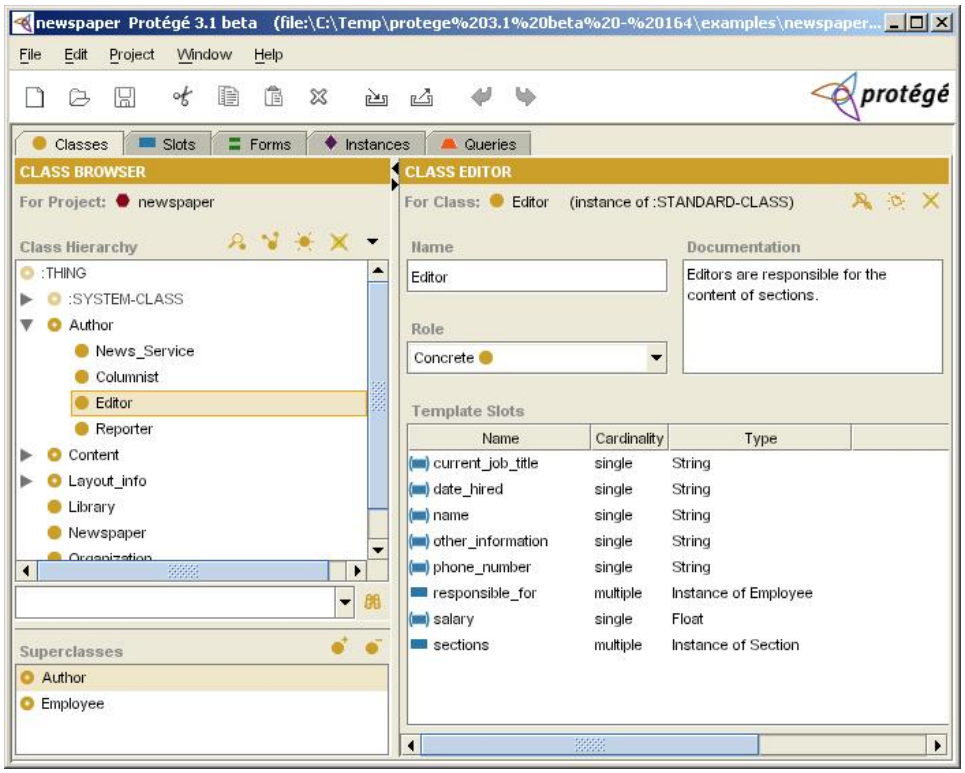


Figura 3.11: Pestañas del editor de marcos (captura de pantalla).

- La pestaña de clases (Classes) de la figura 3.11 es un editor de ontologías que se puede utilizar para definir clases y jerarquías de clases, ranuras y restricciones del valor de ranuras, relaciones entre las clases y propiedades de estas relaciones.
- La pestaña formularios (Forms) de la figura 3.11 genera un formulario predeterminado para la introducción de instancias basada en los tipos de ranuras que se han especificado. Se puede cambiar el formulario predeterminado mediante la reordenación de los campos de pantalla, cambio de tamaño, etiquetas y otras propiedades de las ranuras.

- La pestaña de instancias (Instances) de la figura 3.11 es una herramienta de adquisición de conocimiento que se puede utilizar para introducir las instancias de las clases definidas en una ontología.

- La apariencia que ofrece el Editor de OWL es la que aparece en la figura 3.12

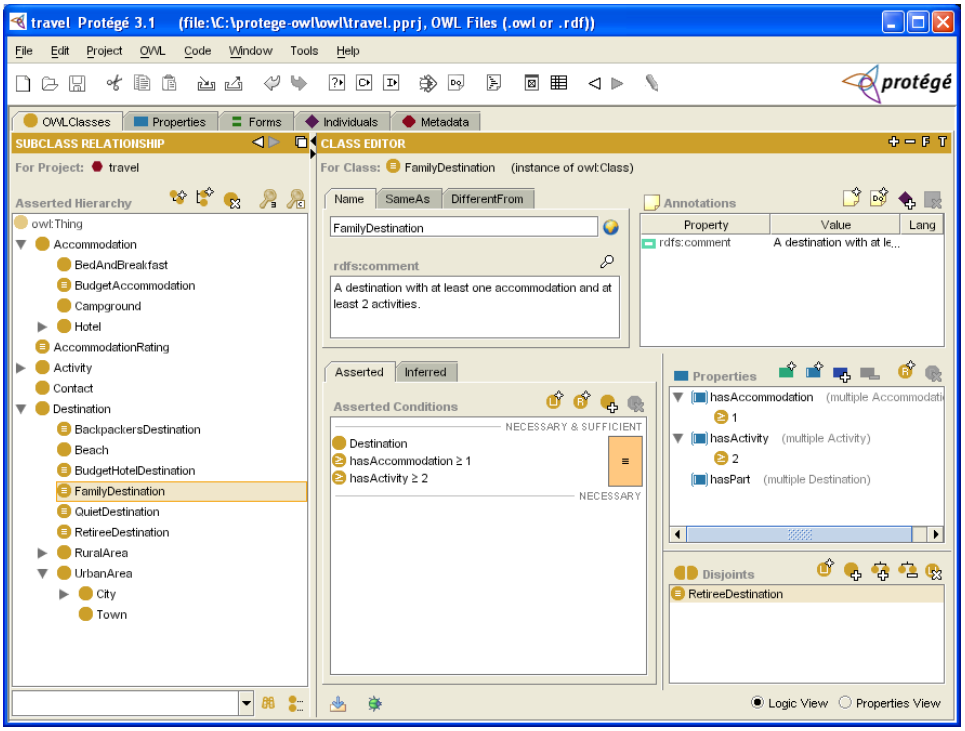


Figura 3.12: Pestañas del editor de OWL (captura de pantalla).

- La pestaña "OWLClasses" de la figura 3.12 se utiliza para editar las jerarquías de conceptos. En la parte derecha de la pantalla se muestran los detalles de la clase seleccionada. La parte superior de esta zona permite a los usuarios añadir comentarios, etiquetas y otras anotaciones. La parte inferior muestra las características lógicas de la clase seleccionada. En Protégé se han integrado a la perfección herramientas de clasificación. Estas herramientas se pueden utilizar para comprobar inconsistencias y relaciones entre las clases y los individuos. Los resultados de la clasificación se muestran en la pestaña OWLClasses, y puede ser fácil de navegar y analizar como se muestra en la imagen.
- La pestaña propiedades (Properties) de la figura 3.12 se utiliza para editar las características de las propiedades en el modelo.

- La pestaña de individuos (Individuals) de la figura 3.12 se utiliza para introducir los datos de la instancia. Los formularios que aparecen en la mitad derecha de la pantalla se generan automáticamente a partir de la definición de clase. Por ejemplo, si una clase tiene una propiedad de tipo *"xsd:string"*, entonces el sistema automáticamente muestra un objeto visual de campo de texto para introducir cadenas.
- El editor de OWL también se puede utilizar para modificar los modelos RDF Schema. La interfaz de usuario se ajusta al lenguaje seleccionado y ajusta la pantalla para presentar objetos visuales para *"rdfs:Classes"*.

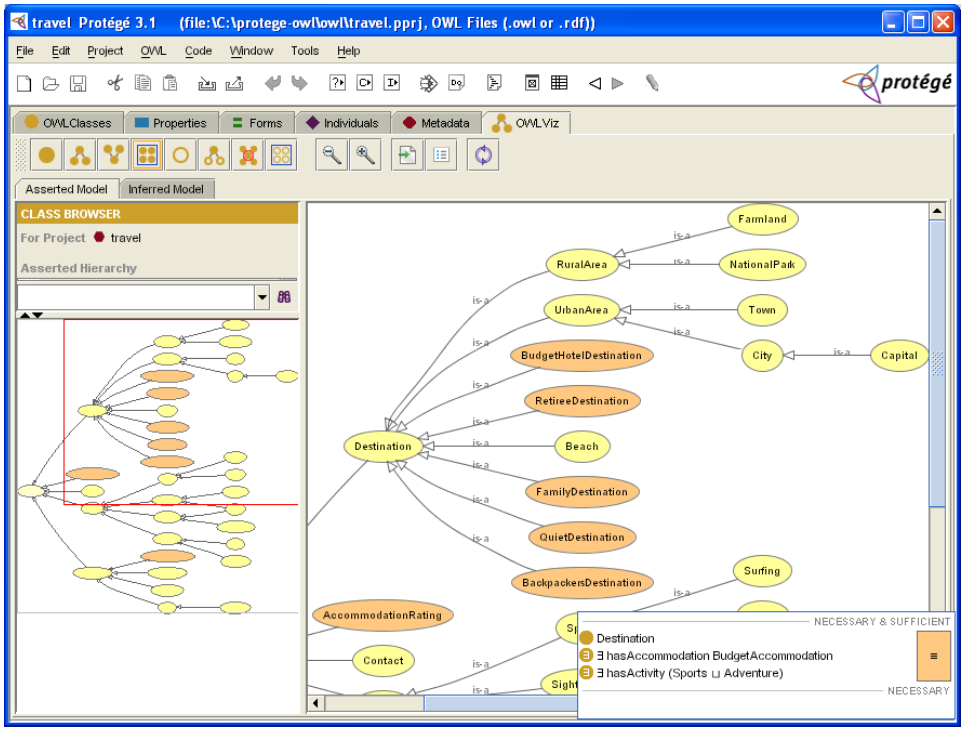


Figura 3.13: Extensión de visualización OWLViz (captura de pantalla).

- La comunidad de desarrolladores de Protégé ha contribuido con numerosas extensiones de la plataforma base. Entre las más populares se encuentra OWLViz, aparece en la figura 3.13, que se utiliza para visualizar ontologías OWL de forma gráfica.
- El editor OWL también soporta la edición de bases de reglas en el lenguaje SWRL (*Semantic Web Rule Language*, Lenguaje de Reglas de la Web Semántica). Las reglas se pueden modificar mediante un editor de expresiones adecuado.

3.4.2.2. pOWL

pOWL[8] (Auer, 2005) es una herramienta de gestión de ontologías que permite el análisis sintáctico, almacenamiento, consulta, manipulación, servicio y serialización de bases de conocimiento OWL en un entorno colaborativo web. Su principal objetivo es servir como banco de pruebas para implementaciones rápidas de nuevos enfoques y, por tanto, los usuarios a los que está dirigida son los investigadores que trabajan en el ámbito de la Web Semántica. También está orientada para ser usada por ingenieros del conocimiento a través de su interfaz web, desde donde se puede desarrollar ontologías de forma colaborativa.

pOWL está desarrollada en el lenguaje web PHP[9] y esta desarrollada bajo licencia de código libre. Se decidió usar PHP como lenguaje de programación por las siguientes ventajas:

- Es independiente de la plataforma, lo cual permite a los desarrolladores e investigadores de la Web Semántica modificar y extender la herramienta pOWL a sus necesidades.
- El código fuente resultante es corto y se intentó que fuera fácil de entender.
- Es el lenguaje de programación web más utilizado por las aplicaciones web, se estima que el 35 % de los portales web disponen de él y, por lo tanto, supera a todas las demás tecnologías de aplicaciones web.
- Dada la difusión de PHP, para que el paradigma de la Web Semántica tenga éxito, necesariamente ha de disponer de herramientas y aplicaciones que interaccionen con dicho lenguaje.

La principal razón que estuvo detrás del desarrollo de la herramienta pOWL fue que no existía en su momento ninguna herramienta que permitieran conjuntamente el trabajo colaborativo, la interfaz de usuario web y el manejo nativo de RDFS y OWL. Así lo demuestra el estudio de 94 editores de ontologías realizado por Denny (2004).

En definitiva, pOWL intenta proporcionar a la comunidad Open-Source una solución de gestión de ontologías Web, fácil de instalar, fácil de usar y escalable, que cubra el ciclo de vida completo de las ontologías. Algunas de las características más sobresalientes de pOWL son:

- Soporta la observación y edición de ontologías RDFS/OWL de cualquier tamaño.

[8] `http://powl.sourceforge.net/`
[9] `http://www.php.net`

- Dispone de objetos visuales sofisticados para la edición de datos.
- Diseñado según filosofía Plugin. pOWL es fácilmente extensible.
- Sistema de consulta a la base de conocimiento. pOWL ofrece un constructor de consultas en RDQL así como búsqueda de texto completo para literales y recursos.
- Dispone de un esquema de autenticación. Los privilegios de los usuarios y grupos están dirigidos a ser asignados a modelos, clases y propiedades.
- Escalable y rápido. Los modelos se almacenan en tablas de una base de datos y solo parte del modelo son cargadas en memoria principal.
- Soporte muti-lenguaje.

La apariencia que ofrece la herramienta pOWL es la que se muestra en la figura 3.14. Se puede apreciar cómo la interfaz de usuario está dispuesta en pestañas, cada una representa una visión diferente sobre la base de conocimientos. Las pestañas de las que dispone POWL son las siguientes:

- Modelos (Models): Ofrece una visión general de los modelos que almacena pOWL.
- Tripletas (Triples): Muestra una lista con las tripletas de la ontología que se haya seleccionado donde se puede navegar y realizar búsquedas.
- Clases (Classes): Organiza jerárquicamente las clases y permite la visualización y edición de sus definiciones.
- Propiedades (Properties): Organiza jerárquicamente las propiedades y permite la visualización y edición de sus definiciones.
- Instancias (Instances): Ofrece varios puntos de vista sobre las instancias de clase en un modelo.
- RDQL: Es una implementación de SQL como lenguaje de consulta para RDF. Se diferencia de SparQL en que no proporciona un grafo RDF como resultado de una consulta.
- Búsqueda (Search): Realiza búsqueda de literales y/o IRIs en la base de conocimiento.
- Versión (Version): Permite el acceso a la información sobre la evolución de la ontología

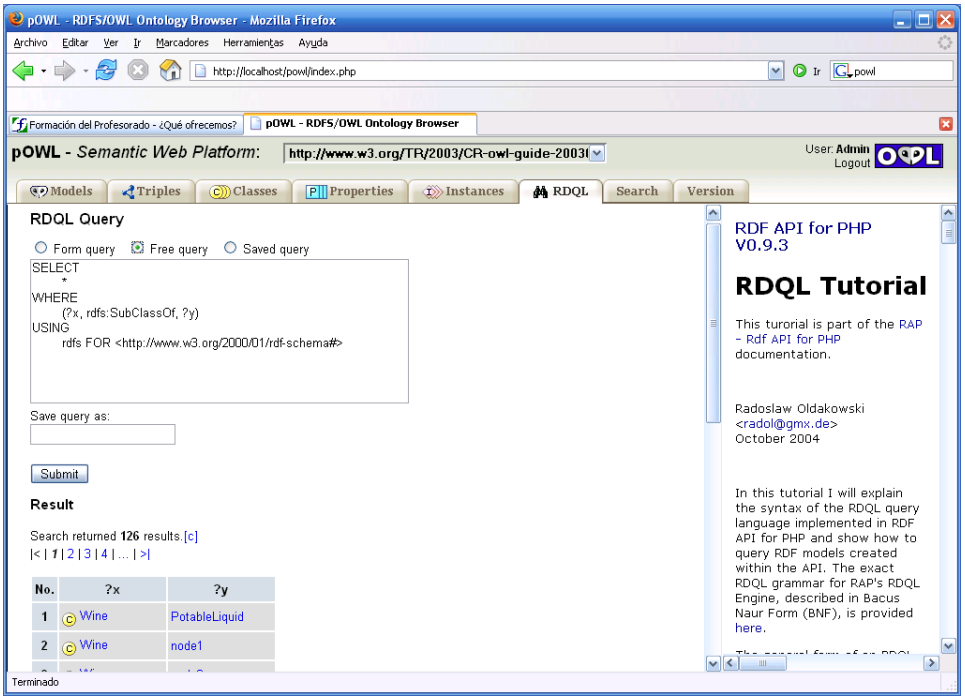


Figura 3.14: Interfaz web de trabajo del entorno pOWL (captura de pantalla).

3.4.2.3. APIs de programación

Este tipo de herramientas ofrecen Interfaces de Programación de Aplicaciones, también conocidas como APIs, que proporcionan a los programadores una infraestructura técnica que les permita desarrollar aplicaciones reales de la Web Semántica, aislándolos de las particularidades de la sintaxis concreta y proporcionándoles una perspectiva de alto nivel de los objetos de una ontología OWL: Clases, propiedades y axiomas.

Herramientas tales como Kaon, Jena y pOWL disponen de APIs de programación, las cuales ofrecen una funcionalidad muy variada y cuyos detalles se presentaron en Navarro-Galindo y Samos (2007a).

Cabe destacar la API de pOWL, dado que se trata de la API que se ha usado en los desarrollos realizados para la presente tesis. La decisión de usar esta API y no otra viene condicionada por la decisión de usar Joomla como CMS donde realizar nuestro trabajo. Dado que Joomla está programado en lenguaje script PHP, se necesitaba una API en este mismo lenguaje que fuese capaz de proporcionar la infraestructura técnica y las herramientas que se requieren a la hora de desarrollar aplicaciones de la Web Semántica. La API

de pOWL cubre éstos requisitos y por este motivo ha sido la elegida.

Los componentes que forman la arquitectura de la API de pOWL están dispuestos de forma apilada y son los siguientes:

- **Powl Store** - El almacenamiento de pOWL se puede realizar mediante cualquier base de datos relacional que soporte ADOdb [10] como capa de abstracción que proporciona el acceso de base de datos mediante el lenguaje script PHP. El diseño de tablas que usa pOWL para almacenar la información relativa a ontologías y a su evolución es el que muestra la siguiente tabla:

Tabla 3.1: Esquema de Base de Datos de Powl Store.

Tabla	**Descripción**
Models	Proporciona información sobre los modelos del almacén
Statements	Contiene las tripletas de un modelo
Log_ actions	Mantiene información sobre las acciones de edición sobre un modelo
Log_ statements	Contiene las sentencias modificadas en cada acción

- **RDFAPI, RDFSAPI, OWLAPI** - APIs dispuestas en capas para el manejo de los formatos RDF, RDFS y OWL.
  - **RDFAPI** es un proyecto independiente de Oldakowski et al. (2004). Dota a pOWL de la siguiente funcionalidad: Parser, serialización de RDF en diferentes formatos, consultas en RDQL, clases y métodos para trabajar con modelos RDF compuestos por recursos y literales, y una API llamada NetAPI para la publicación de modelos en la Web.
  - **RDFSAPI** extiende el esquema de clases de RDFAPI a través de las clases específicas de RDFS.
  - **OWLAPI** extiende las clases de RDFSAPI con los métodos para manejar las propiedades predefinidas de OWL, los axiomas y restricciones DL, así como la realización de una inferencia básica de subsunción.
- **Powl API** - Clases y funciones de alto nivel de abstracción (en la cima de las componentes anteriores) a partir de las cuales construir aplicaciones web.

[10] http://adodb.sourceforge.net/

3.4.2.4. Buscadores y repositorios de ontologías

Se trata de herramientas online que ofrecen servicios de consulta, recuperación y organización de ontologías. Cabe destacar las siguientes:

- **SchemaWeb**[11] se ha convertido en uno de los mayores repositorios de esquemas de metadatos y ontologías. Contiene un buscador de esquemas y un navegador que muestra un directorio de los distintos esquemas existentes que da acceso directo a cada uno ellos. También permite el registro de nuevos esquemas a través de la Web.

- **Swoogle**[12] es un sistema de indexación, recuperación y organización de información para documentos de la Web Semántica, lo que se denomina en Swoogle (SWDs, Semantic Web Documents), o lo que es lo mismo documentos escritos básicamente en RDF y OWL, aunque también DAML en algunos casos. Este motor recupera, procesa, analiza e indexa documentos de la Web Semántica que estén disponibles online, todo ello lo hace a través de un sistema de búsqueda y resultados de interfaz web similar a Google.

3.5. Conclusiones

Se ha avanzado mucho desde que Tim Berners-Lee escribiera el artículo "The Semantic Web" en 2001 (Berners-Lee et al., 2001), sobre todo en los estándares, infraestructura y herramientas necesarias para su despliegue. También se han desarrollado multitud de proyectos y experiencias para poner a prueba todas estas herramientas e ideas. En este punto, un elemento esencial es el desarrollo de aplicaciones reales basadas en la estas tecnologías como hito para que la Web Semántica prospere.

Existe un gran interés en el entorno académico, sector público y empresarial por hacer de la Web Semántica una realidad. Se piensa que las innovaciones que introduce la Web Semántica pueden ser clave para el avance de la sociedad de la información.

Las grandes entidades públicas tales como el programa marco EU-IST en Europa o DARPA en EEUU incluyen áreas específicas dedicadas a la Web Semántica y están disponiendo de grandes presupuestos en proyectos de I+D en este área. En esta misma línea, las principales empresas tales como IBM, Microsoft, Sun, Oracle, BEA, etc. están participando activamente en el desarrollo de los estándares y tecnologías.

[11] `http://www.schemaWeb.info`
[12] `http://swoogle.umbc.edu`

En cuanto al contexto académico, la Web Semántica sigue siendo un tema de moda en las universidades de todo el mundo, entre las que cabe destacar: Stanford, el MIT, Innsbruck, Maryland, Manchester, Karlsruhe o la Open University, entre otras.

En pocos años se ha formado y consolidado una gran comunidad investigadora, cuyo reflejo se traduce en los congresos internacionales específicos para este área como *"Internacional Semantic Web Conference"* o *"Semantic Technology Conference"* y en la gran cantidad de congresos que incluyen la temática de la Web Semántica en sus programas. También cabe destacar importantes revistas que han surgido en este contexto como el *"Journal of Web Semantics"* o el área *"The Semantic Web of Electronic Transactions on Artificial Intelligence (ETAI)"*.

Es de gran importancia el apoyo del W3C en el proyecto de la Web Semántica, con la creación de activos grupos de trabajo para su desarrollo, y especialmente su liderazgo en el esfuerzo de estandarización de lenguajes y tecnologías específicas para la Web Semántica.

Como conclusión, afirmar que los resultados alcanzados a día de hoy en la adopción universal de la Web Semántica son aún preliminares, solamente se han producido los primeros avances y que la gran revolución está aún por llegar. Se necesita desarrollar aplicaciones reales que pongan en práctica los principios de la Web Semántica, que pueblen la Web con metadatos y ontologías, de forma que ésta adquiera la masa crítica imprescindible para ser una realidad. En espera de este objetivo siguen abiertos numerosos campos de investigación e innovación, suficientemente interesantes para motivar la investigación en el área. En este sentido se ha desarrollado la presente tesis.

Capítulo 4

Anotaciones Semánticas en Documentos Web

Confía siempre en el tiempo. Suele dar dulces salidas a muchas amargas dificultades.

Miguel de Cervantes Saavedra

Resumen: La anotación semántica de documentos es el primer paso para permitir el procesamiento automático de la información de la Web, y por tanto para la creación de la Web Semántica. En el capítulo que sigue se estudia lo que se entiende por anotación semántica, los diferentes tipos que existen, los lenguajes de anotación que se usan para ello y las tecnologías de marcado que se usan para delimitar de alguna manera los textos objeto de las anotaciones.

4.1. Introducción

Uno de los aspectos más importantes a la hora de progresar hacia la Web Semántica es cómo convertir el contenido web (el nuevo y el existente) comprensible por las personas en su equivalente semánticamente enriquecido, de manera que sea comprensible por los ordenadores.

El marcado semántico de documentos web es el primer paso hacia la adaptación del contenido web a la Web Semántica. El enriquecimiento semántico se hace posible gracias al marcado de contenido web mediante metadatos, los cuales posibilitan que se describan las entidades que se encuentran en el

contenido y las relaciones entre ellas (Sheth et al., 2002). Proporcionando un significado bien definido a los elementos que actualmente componen la Web se posibilitaría, entre otras cosas, mejorar la capacidad de búsqueda contextual, incrementar la interoperabilidad entre sistemas en un entorno colaborativo y, cuando se combinan con servicios Web, componer aplicaciones automáticamente a partir de los servicios Web publicados (Tsai et al., 2003).

En el diccionario de la Real Academia de la Lengua Española se define "*anotación*" como *"Acción y efecto de anotar"*. Asimismo, se define el término "*anotar*" como *"Poner notas a un escrito, una cuenta o un libro"*.

En el contexto computacional, una anotación consiste en asignar una nota a una porción de texto o a un objeto (imagen, video). Más específicamente, en el contexto de la Web Semántica, la nota asignada contiene información semántica en forma de metadatos con el objetivo de establecer un enlace entre una ontología de referencia (Maedche et al., 2003) y la parte específica del texto u objeto que está siendo marcada. En el caso de texto, es interesante aclarar las diferencias existentes entre los conceptos de anotación y marcado. Las anotaciones semánticas llevan implícito un proceso de análisis, extracción y marcado de la información para enriquecerla semánticamente. El marcado como tal consiste simplemente en delimitar e identificar de alguna manera un fragmento de texto para que posteriormente se le pueda asociar información semántica en forma de metadatos. En ocasiones se usa el término "*marcado semántico*" con el mismo significado que "*anotación semántica*", es decir, se presupone que ya se ha realizado proceso de delimitación e identificación de un fragmento de texto y la asociación de los metadatos oportunos.

4.2. Tipos de Anotaciones Semánticas

Aunque se pueden realizar diferentes clasificaciones atendiendo a diversos criterios, los intereses de esta tesis hace que nos fijemos en los mismos criterios que autores como Popov et al. (2004); éstos son: Localización donde se almacenan y grado de automatización.

De acuerdo al lugar donde se almacenan, tenemos los siguientes tipos de anotaciones:

- **Internas o Embebidas:** Creados con lenguajes de marcado, como por ejemplo RDFa, y almacenadas dentro del mismo documento web donde se realiza la anotación.
- **Externas:** Almacenadas en ficheros o en servicios distintos del documento web que se anota. El acceso a la información anotada se consigue

usando lenguajes de marcado tales como XML o RDF.

Los dos tipos de anotaciones anteriores no son excluyentes, existe la posibilidad de usar un almacenamiento dual de anotaciones, dependiendo del sistema que se pretenda desarrollar, almacenando, por un lado, las anotaciones embebidas en el documento donde se anota (por ejemplo haciendo uso de microformatos o RDFa); y por otro, almacenando las anotaciones en el lado servidor (usando lenguajes ontológicos tales como RDF u OWL). La principal ventaja que plantea el almacenamiento dual es que evita el problema conocido como "La Web profunda" (Bergman, 2001) para las anotaciones en los documentos, por lo que los indexadores de los motores de búsqueda Web podrían acceder a la información semántica de la anotaciones y por otro lado, en el servidor, se podrían usar herramientas de inferencia para chequear la integridad de la información semántica y para deducir nuevo conocimiento.

Conforme a su nivel de automatización, tenemos los siguientes tipos de anotaciones:

- **Directa o Manual:** El usuario realiza las anotaciones directamente en un contenido dado mediante el uso de herramientas específicas. No existe grado de automatización; el usuario es responsable de analizar los documentos, intentado identificar las entidades semánticas y las relaciones entre ellas.
- **Automatizada:** Un proceso automático o semi-automático genera las anotaciones. De alguna manera (Giovannetti et al., 2008), el proceso identifica las entidades semánticas y sus relaciones en el contenido fuente y realiza correspondencias (mapeos) con las entidades y relaciones de ontologías. Por lo general, estas anotaciones se crean con lenguaje de marcas RDF y se almacenan independientemente en ficheros o en una base de datos específica.

sectionLenguajes para la Anotación Semántica Un lenguaje de anotación es un conjunto de marcas semánticas y reglas sintácticas que se usan para describir a un computador la estructura de un documento digital con objeto de representar su significado.

En la figura 4.1 se representa la brecha semántica existente entre la información que el navegador tiene y la que comprendemos los humanos acerca de un mismo documento web. Los lenguajes de anotación trabajan en esta línea, haciendo explícita la información semántica que de manera implícita poseen los contenidos de los distintos documentos web para que los ordenadores puedan trabajar con ella.

En esta sección se estudian los lenguajes estándar de la W3C para el anotado semántico: RDF (Manola y Miller, 2004) y RDFa (Adida y Birbeck,

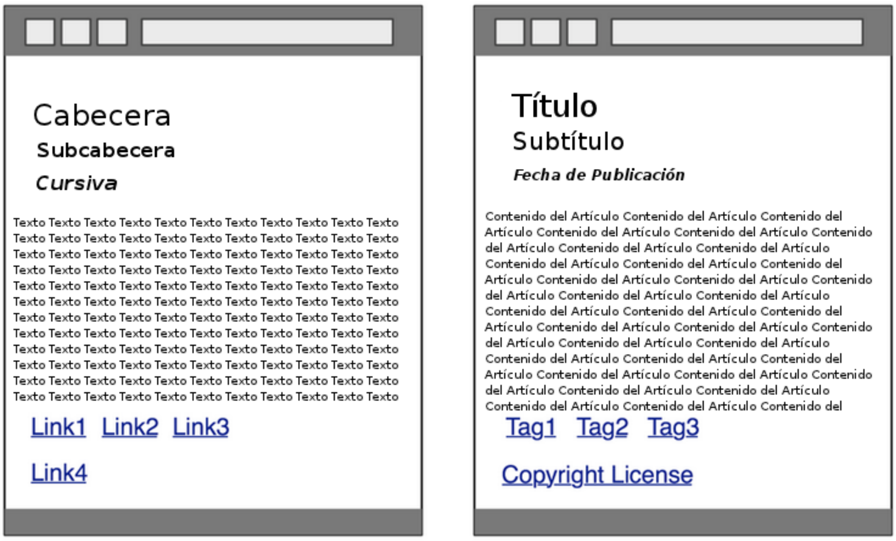


Figura 4.1: Ejemplo de brecha semántica. Imagen extraída de `http://www.w3.org/TR/xhtml-rdfa-primer/`.

2007); así como también la posibilidad de extender la expresividad de los mismos mediante la inclusión de microformatos (vocabularios).

4.2.1. RDF/RDFa

RDF es el marco de trabajo desarrollado en 1997 por la W3C como infraestructura estándar para describir recursos y proporcionar una base para el procesamiento de datos y para posibilitar la interoperabilidad semántica entre aplicaciones Web. Es, por tanto, un estándar para expresar metadatos, que a su vez proporciona una base para hacer uso de los metadatos, así como razonamiento con los mismos.

RDF trata sobre sentencias y tripletas. Existen diferentes sintaxis que pueden ser usadas para expresar dichas tripletas, tales como N3, RDF/XML o RDF. La presente tesis se centra en particular en la sintaxis XML para la codificación de los metadatos, y en la sintaxis RDFa para incrustar RDF dentro de documentos HTML.

RDFa es una sintaxis propuesta por la W3C para expresar datos estructurados RDF en documentos HTML o, lo que es lo mismo, para incrustar semántica en los documentos.

RDFa permite a los autores de contenido HTML marcar datos inteligibles

por humanos con indicadores inteligibles por las máquinas de forma que los navegadores y otros programas los puedan interpretar.

Los atributos "*meta*" y "*link*" de los elementos HTML son usados por RDFa y generalizados de forma que se puedan usar en cualquier elemento de un documento. De acuerdo con la especificación de su sintaxis establecida en el documento `http://www.w3.org/TR/rdfa-syntax/`, los atributos usados son:

- **typeof**: Indica el tipo de instancia descrita.
- **about**: Indica la URI del recurso que describen los metadatos y se refiere al documento actual por defecto.
- **rel, rev, href y resource**: Atributos que establecen una relación o relación inversa con otro recurso.
- **property**: Proporciona una propiedad sobre el contenido de un elemento.
- **content**: Atributo opcional que se sobrepone al contenido del elemento cuando se usa el atributo property
- **datatype**: Atributo opcional que indica el tipo de datos del contenido

Suponiendo que la IRI del documento del ejemplo 4.1 siguiente es *http://example.com/bob*, se van a realizar unas anotaciones semánticas sobre él que permitirán estudiar las ventajas e inconvenientes de realizar anotaciones con los lenguajes RDF y RDFa.

```
<div>
  <h2> The trouble with Bob </h2>
  <div>
    <p> The trouble with Bob is that he takes much better
       photos than I do: </p>
    <img src="http://example.com/bob/photos/sunset.jpg" />
    <p> Beautiful Sunset by Bob </p>
  </div>
</div>
```

Ejemplo 4.1: Documento HTML objeto de anotación semántica.

A continuación, se realizan una anotaciones semánticas sobre el documento del ejemplo 4.1 donde se indica que el título del documento es *"The trouble with Bob"*, se describe que aparece una fotografía titulada *"Beautiful Sunset"* y que su creador es *"Bob"*.

```
<?xml version="1.0" encoding="utf-8"?>
<r:RDF
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:r="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <r:Description r:about="http://example.com/bob/photos/sunset
      .jpg">
    <dc:title>Beautiful Sunset</dc:title>
    <dc:creator>Bob</dc:creator>
  </r:Description>
  <r:Description r:about="http://example.com/bob">
    <dc:title>The trouble with Bob</dc:title>
  </r:Description>
</r:RDF>
```

Ejemplo 4.2: Anotaciones en RDF.

El ejemplo 4.2 muestra cómo se realizan las anotaciones semánticas comentadas anteriormente utilizando RDF para ello. En la práctica, esta información estaría almacenada dentro de un fichero plano de texto o se generaría mediante una página web dinámica; además necesitaría ser enlazada dentro de la cabecera (elemento *"<head>"*) del documento HTML del cual se describen los recursos mediante un elemento *"<link>"* tal y como se muestra en el ejemplo 4.3. Este enfoque es el que descrito por Beckett (2004b) en su especificación de la sintaxis RDF/XML, el cual ha sido usado durante varios años por la DCMI (*Dublin Core Metadata Initiative*, Iniciativa de Metadatos Dublin Core) en su web.

```
<head>
   <link rel="meta" type="application/rdf+xml"
         href="http://example.com/bob/bob.rdf" />
</head>
<div>
  <h2>   The trouble with Bob  </h2>
  <div>
    <p> The trouble with Bob is that he takes much better
       photos than I do: </p>
    <img src="http://example.com/bob/photos/sunset.jpg" />
    <p> Beautiful Sunset by Bob </p>
  </div>
</div>
```

Ejemplo 4.3: Vínculo desde HTML a un documento externo RDF.

Se puede observar en los ejemplos 4.2 y 4.3 que los literales e IRIs implicados en una anotación semántica aparecen tanto en el documento web que se anota como en el documento externo RDF con el que está vinculado. Este

hecho presenta dos problemas principalmente:

1. **Duplicidad de la información:** La información puede ser redundante y repetirse en el documento web que se anota y en el documento de anotación RDF.

2. **Inconsistencias:** Va en contra de favorecer la evolución natural de los documentos anotados, es decir, cuando se modifica el contenido de un documento web también es necesario actualizar el fichero RDF que contiene los metadatos del mismo; en caso contrario, los metadatos de la anotación semántica no estarán en consonancia con la información que ofrece el documento web anotado.

```
<div xmlns:r="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:dc="http://purl.org/dc/elements/1.1/"
         about="http://example.com/bob">
   <h2 property="dc:title">The trouble with Bob</h2>
   The trouble with Bob is that he takes much better photos
      than I do:
   <div about="http://example.com/bob/photos/sunset.jpg">
     <img src="http://example.com/bob/photos/sunset.jpg" />
     <span property="dc:title">Beautiful Sunset</span>
      by
     <span property="dc:creator">Bob</span>
   </div>
</div>
```

Ejemplo 4.4: Anotaciones en RDFa.

Uno de los principales objetivos de RDFa es incrustar RDF dentro de documentos HTML sin repetir contenido cuando ese contenido son datos estructurados. En el ejemplo 4.4 se aprecia cómo se han integrado las anotaciones semánticas RDFa con el documento original, evitando los problemas de duplicidad de contenidos y de inconsistencia entre metadatos y contenidos.

Una de las desventajas de RDFa es que, como cualquier tecnología reciente, pocas herramientas de la Web Semántica la soportan. Por ejemplo, si fuera necesario realizar una consulta sobre el recurso anotado en la Figura 4.4 anterior se usaría naturalmente el lenguaje de consultas SPARQL. Sin embargo, hoy día los motores de búsqueda SPARQL necesitan RDF/XML. Por esto mismo se necesitan procesadores de RDFa tales como RDF Distiller[1], capaces de convertir RDFa en formato RDF/XML.

[1] `http://www.w3.org/2007/08/pyRdfa`

4.2.2. Microformatos

Un microformato es una propuesta para el marcado semántico de documentos web cuyo objetivo también es reusar etiquetas HTML/XHTML para transportar metadatos y otros atributos en páginas web y en otros contextos que soportan HTML/XHTML, así como RSS. Esta propuesta posibilita al software procesar la información inteligible por usuarios finales automáticamente (Khare, 2006).

La premisa que apoya el surgimiento de los microformatos es la siguiente: Las etiquetas de formato tradicionales para mostrar la información en la Web no describen lo que significa la información. Los microformatos pueden llenar este vacío semántico y, por otro lado, obviar otros métodos de tratamiento automatizado más complicados como por ejemplo el procesamiento de lenguaje natural. El uso de microformatos permite que los datos sean indexados, buscados y guardados de forma que la información pueda ser reutilizada o incluso combinada.

En definitiva, un microformato es una porción de código HTML o XHTML estándar cuyo objetivo es insertar contenido semántico en secciones de documentos XHTML o HTML. Para ello, se utilizan los atributos HTML *"class", "id", "rel" y "rev"*. Actualmente existen decenas de microformatos[2], algunos de ellos todavía están en fase de borrador. Algunos de los microformatos más importantes son:

- **hCalendar**: Microformato basado en el estándar iCalendar que describe los eventos de una agenda, como puede ser el resumen del evento, la localización, la hora de inicio y la hora de fin, entre otros.
- **hCard**: Microformato basado en el estándar vCard utilizado para representar personas, organizaciones, compañías y lugares. Define elementos como el nombre, el número de teléfono, la URI de la fotografía, el título y la dirección, entre otros.
- **hReview**: Microformato estándar para la descripción de opiniones o críticas sobre un determinado tema. Entre los elementos que define está el resumen, el lugar, el asunto y la fecha de visita, entre otros.
- **XFN**: Microformato simple dedicado a representar las relaciones humanas por medio de enlaces y la utilización del atributo rel para definir el tipo de relación.
- **RelLicense**: Microformato encaminado a la definición de la licencia de un determinado contenido.

[2] `http://www.microformats.org`

- **RelTag**: Microformato para mostrar una etiqueta o tag. Está formado por un elemento HTML "*a*" con un atributo *"rel="tag""* y un URL, siendo el último elemento del camino de dicha URL la etiqueta, e identificando el sitio en el que se etiqueta como el resto del camino de la URL, que debería ser un espacio de etiquetas (tag space) donde se definen o comparan etiquetas.

- **xFolk**: Microformato en fase de borrador utilizado para definir una Folksonomía aplicada sobre una determinada URL. Está compuesto por el enlace que se etiqueta, el título de dicho enlace, las etiquetas aplicadas a dicho enlace (en formato RelTag) y una descripción extendida del bookmark anotado.

Uno de los principales problemas de añadir semántica a los contenidos web radica en el escaso uso de herramientas que generen metadatos que describan el contenido de forma automática ya que, principalmente, tiene que realizarse todo de forma manual.

```
<div>
  <div>Joe Doe</div>
  <div>The Example Company</div>
  <div>604-555-1234</div>
  <a href="http://example.com/">http://example.com/</a>
</div>
```

Ejemplo 4.5: Información de un contacto en formato HTML.

```
<div class="vcard">
   <div class="fn">Joe Doe</div>
   <div class="org">The Example Company</div>
   <div class="tel">604-555-1234</div>
   <a class="url" href="http://example.com/">http://example.com/</a>
</div>
```

Ejemplo 4.6: Codificación de un contacto usando el microformato hCard.

En el ejemplo 4.6, definido a partir del ejemplo 4.5, los atributos nombre (*fn*), organización (*org*), número de teléfono (*tel*) y la dirección web (*url*) han sido identificados con nombres específicos de clase y todo la información está envuelta en *"class = "vcard""*, lo que indica que las clases forman el microformato hCard y no son simplemente coincidencias en el nombre de las clases. Existen extensiones para los navegadores que pueden extraer la información y transferirla a otras aplicaciones tales como una libreta de direcciones.

4.3. Métodos de Marcado

Se trata de un aspecto importante cuando se llevan a cabo anotaciones semánticas, ya que es necesario delimitar, de algún modo, el fragmento de texto sobre el cual se realiza la anotación.

Un *rango* es una parte arbitraria de un documento HTML, definida por puntos límite que denotan el comienzo y el fin del mismo. Un rango puede comenzar y terminar en cualquier punto, e incluso el comienzo y fin pueden coincidir (rango vacío). El rango más común es la selección de texto que realizan los usuarios y que será muy útil a la hora de realizar anotaciones semánticas sobre los fragmentos de texto que se seleccionen.

```
<div id="e1196">
   <a href="http://ex.com/myblog.html" class="external">
               FLERSA Tool Blog
   </a>
   <p>  FLERSA calls for a Blogger Code of Conduct. Proposals:
      <ol>
        <li>Take responsibility not just for your own words</li>
        <li>Label your tolerance level for abusive comments.</li>
        <li>Consider eliminating anonymous comments.</li>
      </ol>
   </p>
</div>
```

Ejemplo 4.7: Selección básica de texto.

En el ejemplo 4.7 anterior, se muestra con fondo gris un rango de texto que corresponde con el contenido “tolerance level for abusive”.

Esta sección se centra en el estudio de las tecnologías que permiten la creación de rangos, las cuales van a proporcionar la herramienta adecuada para conseguir el objetivo de delimitar e identificar los fragmentos de texto donde se realizarán las anotaciones.

4.3.1. El Problema del Marcado de Contenidos HTML

Aunque la tarea del marcado de fragmentos de texto dentro de un documento HTML pueda parecer simple, en la práctica presenta un importante problema: El texto que compone un fragmento puede atravesar diversas etiquetas HTML y, por tanto, pueden producirse inconsistencias desde el punto de vista del lenguaje de marcas HTML/XHTML, a la hora de crear los rangos que delimitan estos fragmentos de texto.

Para su mejor comprensión, se va a mostrar, en el ejemplo 4.8, un frag-

mento de texto que aparece sombreado, y que se pretende delimitar usando rangos.

```
<div id="e1196">
   <a href="http://ex.com/myblog.html" class="external">
              FLERSA Tool Blog
   </a>
   <p>FLERSA calls for a Blogger Code of Conduct. Proposals:</p>
   <ol>
     <li>Take responsibility not just for your own words, but for the
         comments you allow on your blog.</li>
     <li>Label your tolerance level for abusive comments.</li>
     <li>Consider eliminating anonymous comments.</li>
  </ol>
</div>
```

Ejemplo 4.8: Selección de texto multi-etiqueta.

Sin ninguna intervención, si se crea un rango correspondiente al fragmento de texto sombreado en el ejemplo 4.8, obtenemos el código HTML que muestra el ejemplo 4.9.

```
 calls for a Blogger Code of Conduct. Proposals:</p>
<ol>
     <li>Take responsibility not just for your own words, but for
         the
               comments you allow on your blog.</li>
     <li>Label your tolerance 

```

Ejemplo 4.9: Fragmento de código HTMl seleccionado.

Se puede observar en el ejemplo 4.9 anterior, que si simplemente se copia el texto para formar un rango, el resultado es un fragmento de código HTML no válido. Se han realizado tres marcas sombreadas en los lugares del rango donde se echan en falta etiquetas HTML.

```
<p> calls for a Blogger Code of Conduct. Proposals:</p>
<ol>
     <li>Take responsibility not just for your own words, but for
         the
               comments you allow on your blog.</li>
     <li>Label your tolerance </li>
</ol>
```

Ejemplo 4.10: Fragmento de código HTML válido.

Una API de programación adecuada debe permitir crear rangos capaces de devolver código HTML de tal forma que el fragmento sea válido, como se muestra más arriba en el ejemplo 4.10.

4.3.2. DOM Range

DOM Range (Kesselman et al., 2000) forma parte de la especificación "Document Object Model Level 2 HTML" (Stenback et al., 2003) recomendada por la W3C como plataforma e interfaz de lenguaje que permite a programas y scripts acceder dinámicamente y actualizar el contenido y la estructura de documentos HTML/XHTML.

La tecnología DOM Range ayuda en el proceso de marcado ya que permite la definición de rangos mediante el recorrido e identificación de rangos de contenido en un documento.

Actualmente, los navegadores modernos -como Firefox, Safari y Opera- disponen de una API para documentos XML y HTML donde implementan la especificación DOM Range, permitiendo a los desarrolladores modificar el contenido y la presentación visual de documentos mediante el uso de la misma en lenguaje *javascript*. En particular, Mozilla [3] dispone de un objeto llamado *Range* que es capaz de tratar correctamente un fragmento de un documento como un árbol DOM.

Mediante el uso de objetos del tipo Range, el desarrollador puede encontrar los puntos de inicio y de fin de una selección de usuario dentro de un navegador web, definiendo rangos en los cuales anotar, hacer copias, borrar, reemplazar un rango con otro y otras funciones. Las funcionalidades ofrecidas son las que siguen:

- Creación de Rangos.
- Cambio en la posición de un Rango.
- Comparación de puntos límite entre Rangos.
- Borrado del contenido de un Rango.
- Extracción del contenido.
- Clonado del contenido.
- Inserción de contenido.
- Rodear un contenido.
- Modificar un Rango en base a modificaciones e inserciones.

[3] https://developer.mozilla.org/en/DOM/range

4.3.3. TextRange

La API de Microsoft llamada MSHTML/DHTML es el principal componente del navegador web Internet Explorer. Se puede usar para proporcionar funcionalidad de navegador a cualquier aplicación. También se puede usar la API MSHTML/DHTML para acceder al modelo de objetos subyacente a HTML y modificar cualquiera de sus elementos de forma dinámica (DHTML).

La API MSHTML/DHTML también está disponible para ser usada desde el lenguaje *javascript* y proporciona un objeto llamado *"TextRange"*[4], el cual representa a un fragmento de texto dentro de un elemento HTML. TextRange dispone de multitud de propiedades y métodos similares a los que marca especificación DOM Range de la W3C, aunque no respeta en absoluto la interfaz que se establece en dicha especificación y resulta muy laborioso conseguir la misma funcionalidad usando los métodos que provee. Este problema está presente en el navegador Internet Explorer para versiones inferiores a la 9.

La interfaz DOM Level 2 recomendada por la W3C se ha implementado en las nuevas APIs con las que cuenta el navegador Internet Explorer versión 9 [5]. En la actualidad, Internet Explorer 9 proporciona un objeto de programación para el lenguaje javascript llamado *"HTMLDOMRange Object"* [6] que cumple a la perfección con la especificación DOM Range de la W3C y que, por tanto, cuenta con las mismas funcionalidades que el objeto *"Range"* que ofrece la API de Mozilla descrito en la sección 4.3.2 anterior.

4.3.4. XPointer

XPointer (DeRose et al., 2001) es una tecnología estándar de la W3C que proporciona un mecanismo formal para identificar de forma única fragmentos de documentos XML con objeto de crear enlaces. La mayoría de las herramientas de anotación tienden a usar la tecnología XPointer así como patrones y expresiones regulares.

La especificación XPointer proporciona una forma de identificar posiciones dentro de la estructura interna de documentos XML. Esto posibilita que se lleve a cabo un análisis de la estructura interna del documento, junto con una selección de sus elementos internos basada en varios elementos, como son el tipo de elemento, valores de los atributos, caracteres que forman el contenido y posiciones relativas. En particular proporciona referencias espe-

[4] http://msdn.microsoft.com/en-us/library/ie/ms535872(v=vs.85).aspx
[5] http://msdn.microsoft.com/library/ff974378.aspx#_dom
[6] http://msdn.microsoft.com/es-es/library/ff974356.aspx

cíficas para elementos, cadenas de caracteres y otras partes del documento XML, independientemente de que exista o no un atributo identificador (ID).

XPointer se construye sobre XPath (*XML Path Language*) (Berglund et al., 2007). XPath es un lenguaje declarativo recomendado por la W3C que es usado para navegar dentro de un documento XML mediante la selección de elementos, atributos y otros. Se basa en gran medida en expresiones XPath, similares a expresiones regulares, aunque diseñadas siguiendo la estructura jerárquica de XML. La extensión XPointer permite a XPath realizar las siguientes funciones:

- Seleccionar puntos, intervalos y nodos.
- Usar comparación de cadenas para buscar información.
- Usar expresiones de direccionamiento en referencias URI así como identificadores de fragmento.

A continuación vamos a ver un ejemplo de cómo usar la tecnología XPointer para delimitar el fragmento de texto seleccionado por el usuario, el que aparece a continuación en la figura 4.7 con fondo gris, en el cual sería posible realizar una anotación semántica.

XPointer usa la función *"string-range"* para referenciar en general a fragmentos de texto usando la siguiente lógica de funcionamiento: Se indican la ruta de búsqueda, dentro de la jerarquía de nodos del documento XHTML, hasta el nodo que contiene el texto y a continuación se indica la posición inicial donde comienza el fragmento de texto y cuántos caracteres de longitud tiene.

```
1 http://ex.org/sample.html#xpointer(string-range(id("1196")/p[1]/ol[1]/li[2],""
      ,12,27))
```

Ejemplo 4.11: Enlace XPointer.

El ejemplo 4.11 anterior, corresponde a una referencia al fragmento de texto sombreado en el ejemplo 4.7 usando la tecnología XPointer. Se puede observar que el primer argumento de la función *"string-range"* es *"id("1196") /p[1]/ol[1]/li[2]"* e indica que el fragmento de texto se ubica dentro del segundo elemento *"<LI>"* perteneciente al primer elemento *"<OL>"*, del primer párrafo *"<P>"* perteneciente al elemento con identificador *"e1196"* de la jerarquía DOM del documento cuyo contenido figura en el ejemplo 4.7. El segundo argumento es la cadena vacía *""*, y sirve para recuperar el primer trozo de texto disponible en el conjunto de nodos expresados en el primer argumento. Finalmente, en los argumentos tercero y cuarto se puntualiza

que el texto marcado comienza en el carácter con posición 12 y tiene 27 caracteres de longitud.

```
1 http://ex.org/sample.html#xpointer(string-range(id("1196")/p,"",8,54))xpointer(
      string-range(id("1196")/ol[1]/li[1]))xpointer(string-range(id("1196")/ol[1]/
      li[2]),"",1,20)
```

Ejemplo 4.12: Enlace XPointer.

El ejemplo 4.12 muestra cómo se referenciaría a un rango, el correspondiente al fragmento de texto sombreado del ejemplo 4.8, en el caso de que el texto recorra diversas etiquetas HTML, usando referencias *xpointer* encadenadas.

4.4. Herramientas de Anotación Semántica

Este tipo de herramientas han sido diseñadas para permitir a los usuarios insertar y mantener marcas dentro de páginas Web, ya sea de forma manual como (semi)automática, a partir de las cuales enriquecer con metadatos el contenido de las mismas. La mayoría de estas herramientas han aparecido recientemente, junto al surgimiento de la Web Semántica y, algunas de ellas, están siendo integradas en entornos de desarrollo de ontologías.

4.4.0.1. Amaya

Amaya[7] (Quint y Vatton, 1997) es el editor Web, es decir, una herramienta para crear y actualizar documentos directamente en la Web (ver figura 4.2). Las funciones de navegación se integran perfectamente con las funciones de edición y de acceso remoto en un entorno uniforme. La herramienta mantiene la idea de que la Web es un espacio de colaboración y no sólo un medio de publicación.

Amaya fue creado por la W3C in 1996 para proporcionar un marco de trabajo en el cual integrar las tecnologías W3C. Se utiliza para demostrar estas tecnologías en acción al tomar ventaja de su uso combinado en un entorno único y coherente. Amaya comenzó siendo un editor HTML y de hojas de estilo CSS. Desde entonces ha ido extendiendo su funcionalidad hasta soportar XML y un gran número de aplicaciones XML tales como la familia XHTML, MathML y SVG. Permite que todos los vocabularios se editen simultáneamente en documentos compuestos.

[7] `http://www.w3.org/Amaya`

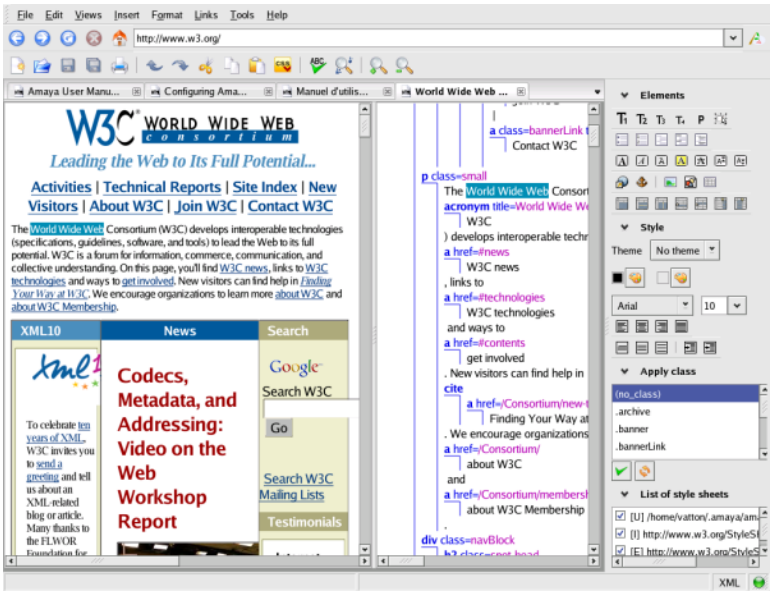


Figura 4.2: Interfaz de la herramienta Amaya (captura de pantalla).

Amaya puede realizar el marcado de contenidos web en documentos HTML o XML, utilizando la tecnología XPointer para determinar el punto de enlace de una anotación en un documento.

4.4.0.2. DBin

DBin[8] (Tummarello y Morbidoni, 2008) es una aplicación de propósito general de la Web Semántica que permite, a usuarios expertos en un dominio, crear "*grupos de discusión*" donde anotar cualquier tema de interés (ver figura 4.3). A bajo nivel, estas anotaciones semánticas se expresan en RDF y el intercambio de información se produce siguiendo un modelo P2P, aunque el usuario final no tiene que ser consciente de ello.

Para un usuario final, DBin es simplemente una manera de expresar y recuperar los conocimientos con otros usuarios demás de una manera más específica de lo que la Web permite.

DBin es relativamente sencillo de usar y de configurar. Los usuarios avanzados crean dichos "grupos de discusión con simples archivos de configuración basados en XML (o clases Java para aplicaciones avanzadas). No se necesita programación para la mayoría de las aplicaciones, pero se pueden programar extensiones de DBin para lograr integraciones específicas con software

[8] `http://www.dbin.org`

existente o con la lógica de negocio. Las principales características que pro-

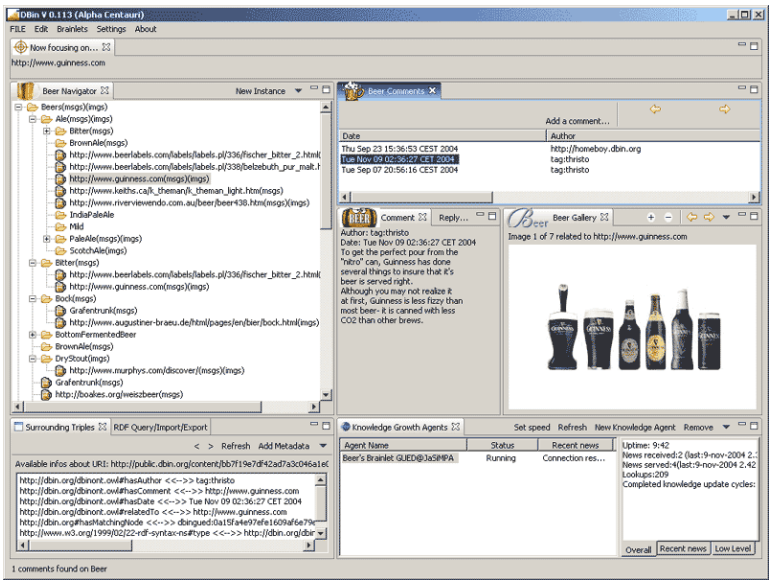


Figura 4.3: Interfaz de la herramienta DBin (captura de pantalla).

porciona la herramienta DBin son:

- Basada en los estándares abiertos de la W3C. Los datos pueden ser exportados y serán interoperables con otras herramientas de la Web Semántica.

- El algoritmo de P2P (RDFGrowth) ha sido diseñado para causar la mínima carga computacional a otros en la red (No hay consultas distribuidas).

- Usa firmas digitales en cada anotación para realizar un seguimiento de su autoría y para que quede claro "quién dijo qué".

- El almacenamiento local permite operaciones de máxima velocidad. Además, el usuario puede aplicar las reglas de filtrado y políticas de confianza a nivel local como considere oportuno.

- El almacenamiento local de los metadatos permite que se puedan integrar en una única vista junto con la información que el usuario tiene a nivel local. El resultado es que se pueden realizar consultas desde una visión integrada de datos P2P y fuentes de datos locales como bases de datos, archivos de escritorio y otros recursos.

- Interfaz sensible, con muchas herramientas integradas que permiten buscar, explorar y editar anotaciones y datos asociados a ellas. La edición de metadatos está guiada por ontologías.
- Dispone de herramientas integradas que ayudan a publicar y recuperar los datos directamente a/desde las cuentas de publicación en la Web.
- Dispone de herramientas para los expertos de dominio con las cuales crear entornos para usuarios finales, llamados *"Brainlets"*, que se encargan de todos los detalles de la ingeniería del conocimiento y proporcionan funcionalidades y herramientas bien enfocadas.
- Licencia de código libre GNU/GPL.
- Basada en la plataforma Eclipse Rich Client Platform para: Conseguir una apariencia independiente del Sistema Operativo, ser multiplataforma, ser extensible mediante un sistema de plugins.

4.4.0.3. GoNTogle

GoNTogle[9] (Bikakis et al., 2010) es una herramienta de anotación y recuperación de documentos, construido en la cima de las tecnologías de la Web Semántica y de Recuperación de Información (tiene como capa subyacente las API que proporciona Apache Lucene[10] y Protégé). GoNTogle permite tanto la realización de anotaciones manuales como automáticas, basadas en ontologías, en documentos con diferentes tipos de formato (doc, pdf, txt, rtf, odt, y sxw). También proporciona herramientas de búsqueda diferentes de la búsqueda tradicional basada en palabras, incluyendo búsqueda basada en ontologías y combinada (ver figura 4.4).

Las principales características que proporciona la herramienta DBin son:

- Las anotaciones se basan en las tecnologías estándar de la Web Semántica, tales como OWL y RDFS.
- Proporciona una forma fácil e intuitiva de anotar documentos usando ontologías.
- Permite a los usuarios abrir y anotar documentos con formatos muy difundidos como pueden ser DOC y PDF, manteniendo su formato original.
- Proporciona un mecanismo de anotación automático basado en la información textual y el historial del usuario, por lo que, ofrece anotaciones personalizadas.

[9] http://web.imis.athena-innovation.gr/projects/gontogle
[10] http://lucene.apache.org/

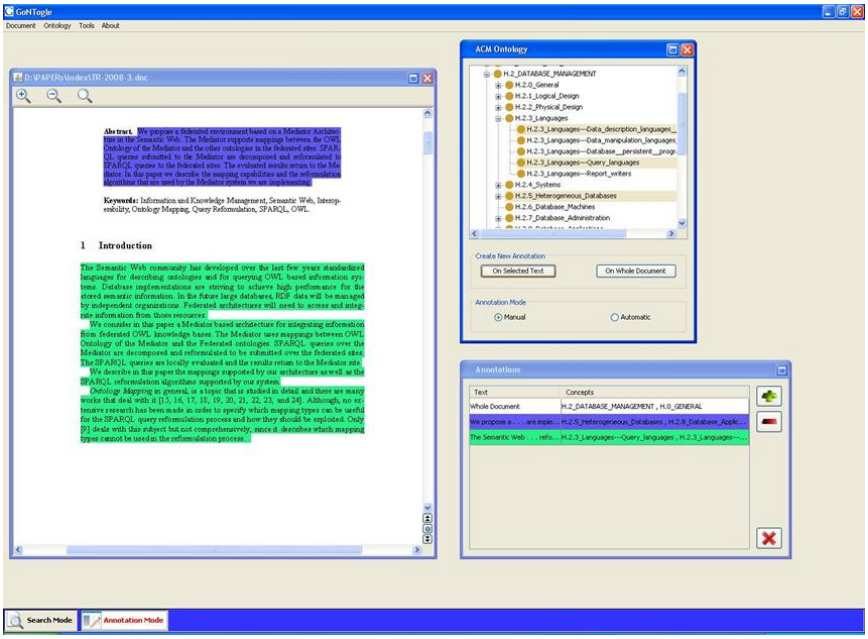


Figura 4.4: Interfaz de la herramienta goNTgle (captura de pantalla).

- Se basa en una arquitectura cliente/servidor, donde todas las anotaciones se almacenan en un sistema centralizado, el cual proporciona un entorno de colaboración donde todas las anotaciones de los documentos son accesibles por todos los usuarios.

- Combina la búsqueda de palabras clave con la búsqueda basada en ontologías, proporcionando herramientas avanzadas de búsqueda para estos dos tipos.

4.4.0.4. KIM

La plataforma KIM[11] proporciona infraestructura y servicios de anotación semántica, indexado y recuperación. La principal diferencia entre KIM y otros sistemas y metodologías es que realiza anotaciones semánticas y proporciona servicios basándose en una ontología y en una masiva base de conocimiento (Popov et al., 2004). En la figura 4.5 se muestra el aspecto de su interfaz.

KIM incluye los siguientes componentes:

- **Proton:** Ontología que contiene sobre 300 clases y 100 propiedades,

[11] `http://www.ontotext.com/kim`

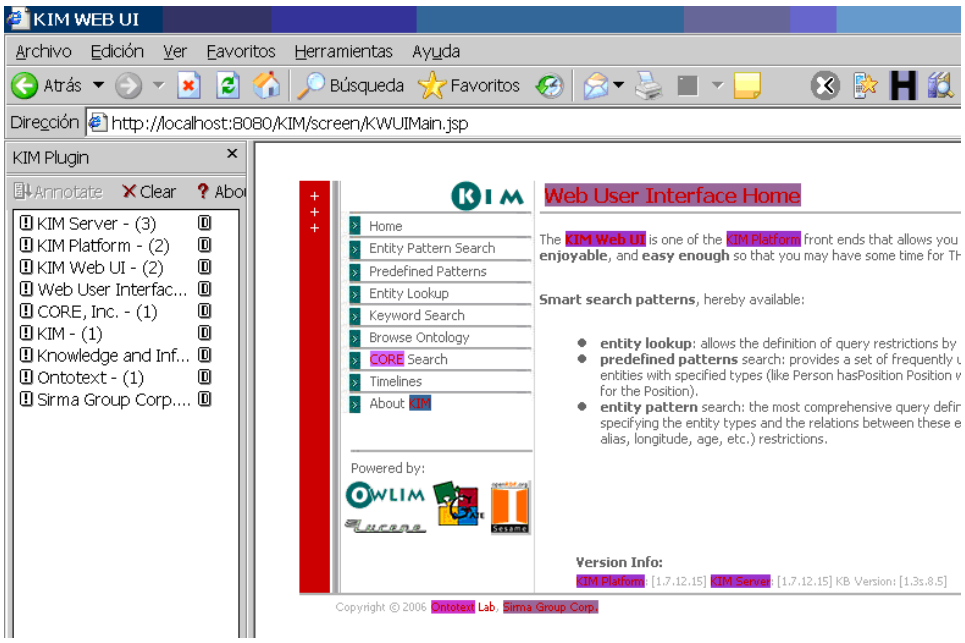


Figura 4.5: Interfaz web de KIM y plugin de marcado (captura de pantalla).

proporcionando cobertura de conceptos generales necesarios para una amplia gama de tareas, como la anotación semántica, el indexado y la recuperación de documentos.

- **Kimso:** Kim System Ontology.
- **Kimlo:** Kim Lexical Ontology.
- **Kim World DB:** Ontología que contiene 250 conceptos y 40 propiedades agrupadas en tres conceptos de nivel superior: Entidad, Entidad Origen y Recurso Léxico.
- **Kim Server**, junto a una API para acceso remoto e integración.
- **Aplicaciones finales (front-ends)**: Kim Web UI y un plugin para Internet Explorer.

Las características mas destacadas de KIM son:

- Anotación semántica de texto: Instanciación automática de una ontología y anotación dinámica en dominios abiertos de contenidos no estructurados y semi-estructurados para la Web Semántica y aplicaciones KIM.

- Indexado y recuperación.
- Consulta y explotación de conocimiento formal.
- Seguimiento de acontecimientos simultáneos y clasificación.
- Análisis de la evolución de población en entidades.

4.4.0.5. MnM

MnM[12] (Vargas-Vera et al., 2002) es una herramienta de anotación que proporciona soporte tanto automatizado y semi-automatizado para la anotación de páginas web con contenidos semánticos.

MnM integra un navegador web, un editor de ontologías y proporciona APIs para conectar con los servidores de ontologías y para integrar herramientas de extracción de información (ver figura 4.6).

MnM adopta un modelo de proceso genérico, de tal forma que pueda ser entendido fácilmente por los desarrolladores web sin que tengan la necesidad de ser ingenieros expertos en ontologías o expertos en tecnologías del lenguaje humano. Otra característica clave de su modelo de proceso es que es genérico con respecto al servidor específico de ontologías y con respecto a las tecnologías de extracción de información utilizadas.

El modelo de proceso de la herramienta MnM se divide en las siguientes actividades:

- **Navegación.** Partiendo de una librería de modelos de conocimiento instalada en un servidor de ontologías, permite elegir un conjunto específico de componentes de conocimiento.
- **Marcado.** Se marca un corpus de documentos donde los componentes de conocimiento seleccionados forman la base del mecanismo de extracción de información.
- **Aprendizaje.** Se ejecuta un algoritmo de aprendizaje para aprender las reglas de extracción.
- **Prueba.** Un mecanismo de extracción de información se ejecuta sobre un corpus de prueba para evaluar su precisión.
- **Extracción.** Se selecciona un mecanismo de extracción de información y se ejecuta sobre un conjunto de documentos.

[12] http://projects.kmi.open.ac.uk/akt/MnM

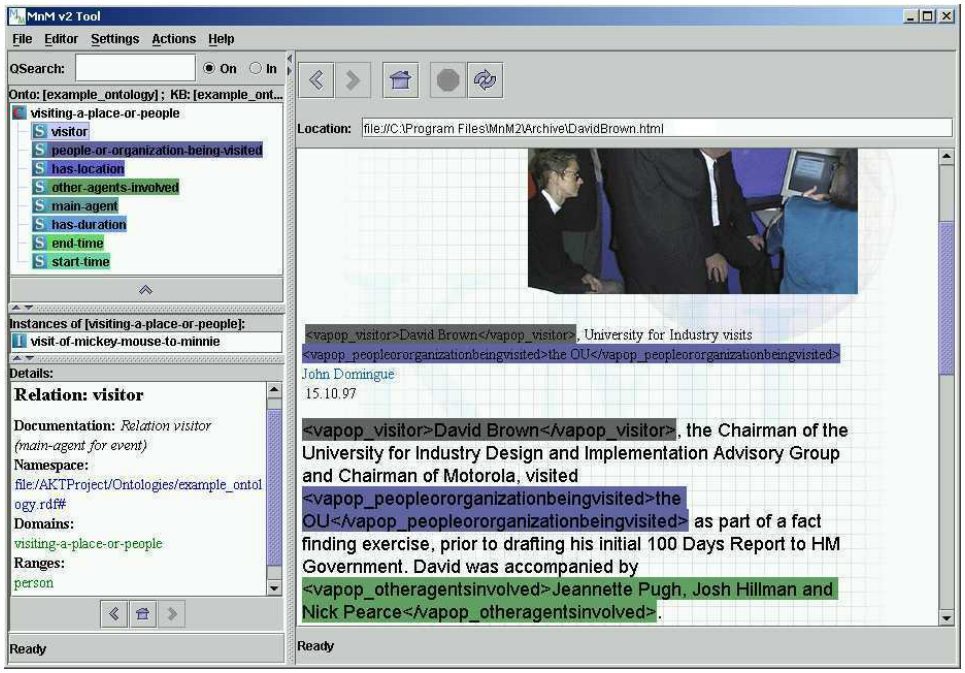


Figura 4.6: Interfaz la herramienta MnM (captura de pantalla).

4.4.0.6. Ontomat

Ontomat[13] (Handschuh et al., 2003b) es una herramienta de anotación de páginas web interactiva y amigable. Ayuda al usuario en la tarea de crear y mantener marcados en OWL basados en ontologías, como por ejemplo para crear instancias, atributos y relaciones. Incluye un navegador de ontologías para explorar la ontología y las instancias de ésta y un navegador HTML que visualiza las partes anotadas del texto. La herramienta está escrita en Java y proporciona una interfaz desde la cual realizar extensiones de la herramienta mediante plugins (ver figura 4.7).

OntoMat dispone de una herramienta de anotación que resalta las partes relevantes de una página web y permite crear nuevas instancias a través de acciones tipo "arrastrar y soltar".

4.4.0.7. Smore

Smore[14] (Kalyanpur et al., 2005) es una herramienta diseñada para permitir a los usuarios el marcado de documentos HTML en OWL utilizando

[13] http://annotation.semanticweb.org/ontomat/index.html
[14] `http://www.mindswap.org/2005/SMORE`

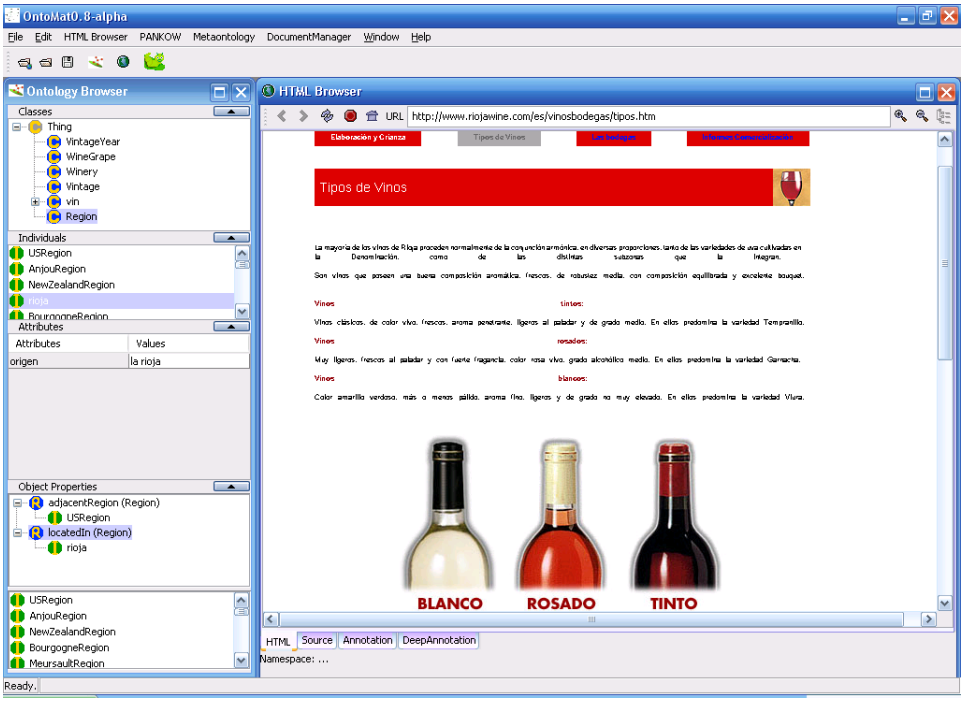


Figura 4.7: Interfaz de Ontomat en forma de navegador de ontologías (captura de pantalla).

ontologías web (ver figura 4.8). Sus objetivos principales son:

- Permitir el marcado de documentos web a usuarios con un conocimiento escaso de los términos y sintaxis de OWL.
- Proporcionar una manera de utilizar las clases, propiedades e individuos de las ontologías existentes, editar una ontología, o incluso crear una nueva ontología dese cero utilizando los términos de los documentos web.
- Proporcionar un ambiente flexible para crear una página web sencilla al mismo tiempo que se realiza su marcado.

Entre las principales características de Smore cabe destacar las siguientes:

- **Centrada en OWL**: Mientras que las versiones anteriores de Smore se basan en herramientas RDF, la nueva versión está diseñada para la fácil creación de ontologías en OWL a partir de la información de la página web. Smore puede importar múltiples ontologías preexistentes en la ontología creada.

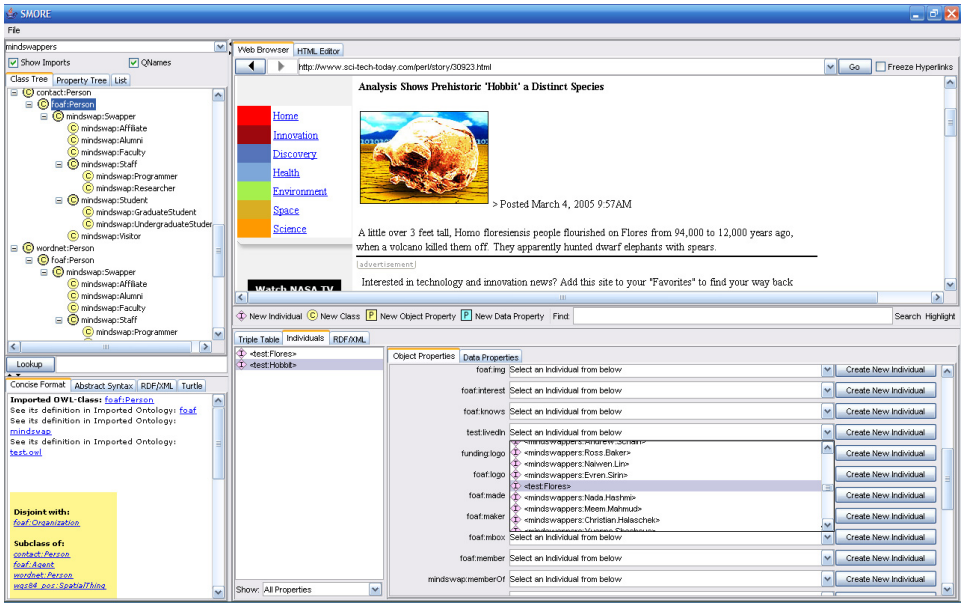


Figura 4.8: Interfaz de Smore con navegador web integrado (captura de pantalla).

- **Creación de Entidades en OWL**: Se pueden crear clases, propiedades e individuos en OWL con sólo seleccionar el texto de cualquier ubicación web o documento cargado y pulsar en el botón correspondiente de la barra de menús. Permite tanto interacción basada en menús como la técnica de arrastrar-y-soltar.

- **Edición de Tripletas en OWL**: Dispone de la capacidad de trabajar con las entidades OWL y con los valores de los datos usando tripletas.

- **Editor Inteligente**: Su editor inteligente utiliza las restricciones de dominio y rango para presentar una lista de objetivos seleccionables para todos los enlaces a propiedades de objetos de un individuo. Para las propiedades de tipo de datos, Smore proporciona un editor adecuado para cada tipo de datos incorporado. Esto hace que la entrada de datos válidos sea simple y elimina problemas con los tipos de datos.

- **Navegador SWOOP**: Integra un navegador de ontologías, proporcionando una forma clara y consistente para buscar y ver clases y propiedades, que además se completa con la funcionalidad de búsqueda.

- **Prevención y resaltado de errores**: Identifica y destaca las tripletas no válidas en la tabla de tripletas, previniendo que la ontología subyacente se convierta en inválida y ayudando al usuario a corregir los errores.

- **Editor HTML Integrado**: Incorpora un editor HTML llamado *EKit*, el cual permite al usuario realizar una edición básica de documentos HTML, conforme están siendo marcados.

4.4.0.8. SOBOLEO

SOBOLEO[15] (Zacharias y Braun, 2007) es una aplicación web para la ingeniería colaborativa de ontologías SKOS y para la anotación de recursos web (ver figura 4.9). Permite a los usuarios crear una taxonomía colaborativamente, utilizarla para anotar los recursos web y utilizar el conocimiento subyacente durante las tareas de búsqueda.

Figura 4.9: Interfaz web de SOBOLEO (captura de pantalla).

Las ontologías SKOS (*Simple Knowledge Organization System*, Sistema de Organización Simple de Conocimiento) (Miles y Bechhofer, 2008) son un modelo de datos común para compartir y enlazar sistemas de organización de conocimiento a través de la Web. El modelo de datos SKOS proporciona un estándar de bajo coste de la ruta de migración para migrar los sistemas de organización del conocimiento a la Web Semántica. SKOS también proporciona un lenguaje ligero e intuitivo para desarrollar y compartir nuevos sistemas de organización del conocimiento. Puede ser utilizado por sí solo o en combinación con lenguajes formales de representación del conocimiento, tales como OWL.

[15] http://tool.soboleo.com

SKOS proporciona un modelo para expresar la estructura básica y el contenido de esquemas de conceptos tales como tesauros, esquemas de clasificación, taxonomías, folksonomías y otros tipos similares de vocabulario controlado. SKOS, como aplicación RDF, permite que la creación de conceptos y su publicación en la Web, enlazados con datos e integrados en otros esquemas conceptuales.

SOBOLEO consta de cuatro partes principales:

- **Búsqueda**: Un motor de búsqueda que busca a través de los recursos de la Web anotados usando la taxonomía como base de conocimiento. Ofrece un buscador para localizar elementos entre las diferentes etiquetas del sistema y entre todo el conocimiento que se ha ido generando.
- **Exploración**: Una interfaz para navegar a través de la taxonomía y de los recursos anotados. Se trata de un área para navegar por la lista de temas, las etiquetas y los recursos del sistema.
- **Anotación**: Una interfaz de anotación para agregar marcadores en el índice. Permite añadir la URL de la herramienta de anotación a los enlaces favoritos del navegador para que funcione a modo de *bookmarklet*, es decir como un marcador que, en lugar de apuntar a una dirección URL, hace referencia a una pequeña porción de código JavaScript para ejecutar tareas de anotación automáticamente. Permite utilizar los conceptos de la taxonomía y etiquetas arbitrarias para anotar los recursos web. Se trata de un área que nos permite enlazar páginas web y personas a los índices de temas. Además nos permite crear nuevas etiquetas y añadir recursos al sistema.
- **Edición**: Un editor colaborativo en tiempo real para la taxonomía. Permite ver los cambios que otras personas hacen a la taxonomía en tiempo real, considerando que se trata de una taxonomía editada colaborativamente por todos los usuarios.

4.4.0.9. Text2Onto

Text2Onto[16] (Cimiano y Volker, 2005) es una herramienta de aprendizaje de ontologías a partir de recursos textuales (ver figura 4.10). Tres son las características principales que distinguen a Text2Onto de su predecesor TextToOnto, así como de otras herramientas dentro del contexto de aprendizaje de ontologías:

1. Consigue la independencia del modelo de la ontología o del lenguaje de representación del conocimiento mediante la representación de los

[16] http://neon-toolkit.org/wiki/1.x/Text2Onto

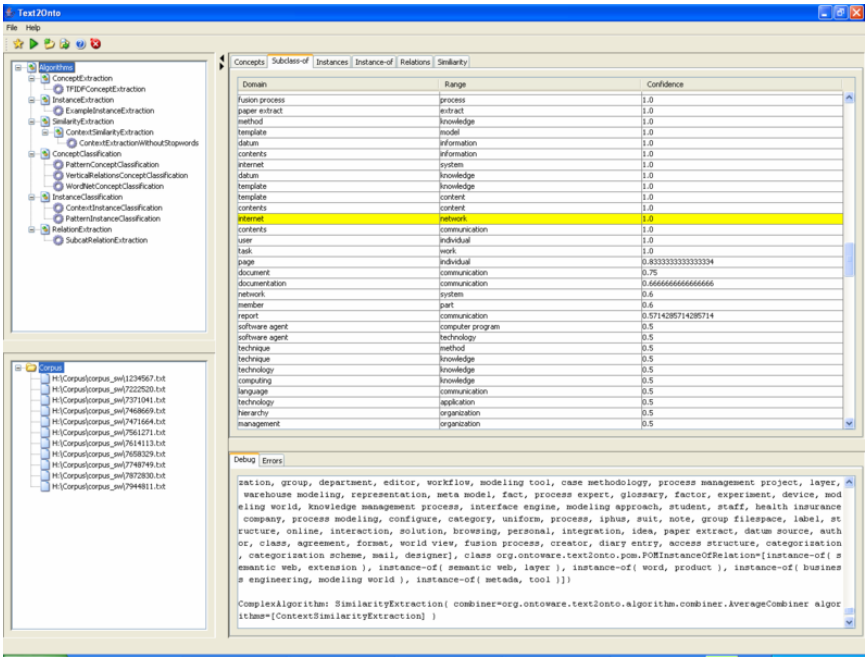


Figura 4.10: Interfaz de usuario de Text2Onto (captura de pantalla).

conocimientos aprendidos en un meta-nivel, en forma de primitivas de modelado instanciadas dentro de un modelo de ontologías probabilísticas llamado POM, se consigue la independencia del lenguaje en favor de la capacidad de traducir las primitivas instanciadas a cualquier formalismo de representación del conocimiento que posea la expresividad suficiente.

2. La interacción del usuario es un aspecto central de Text2Onto y el hecho de que el sistema calcule un nivel de confianza para cada objeto aprendido permite el diseño sofisticado de las visualizaciones de las POM. En definitiva, el hecho de introducir modelos probabilísticos de ontología permite modelos más sofisticados de interacción del usuario.

3. La incorporación de estrategias para el descubrimiento de cambios basados en datos, evita el procesamiento de todo el corpus desde cero cada vez que se produce un cambio, de forma que se produce una actualización selectiva de las POM de acuerdo con los cambios del corpus. Además de aumentarse la eficiencia, estas estrategias permiten a un usuario seguir la evolución de la ontología con respecto a los cambios en el corpus de base (trazabilidad).

4.5. Conclusiones

Este capítulo se ha dedicado a la revisión de los aspectos más relevantes de *anotación semántica en documentos web*.

En la introducción se ha justificado la importancia de la anotación semántica a través de la cual se pueden enriquecer los contenidos de la Web actual con metadatos y avanzar hacia la consecución de la Web Semántica. También se aclara la diferencia entre marcado y anotación, refiriéndose el primer término a la delimitación, de alguna manera, de un fragmento de texto; y el segundo término al hecho de la generación de información semántica en forma de metadatos.

En la sección dedicada a los tipos de anotación se estudian la diversidad existente: Interna o incrustada, externa, manual y automatizada. En principio los diferentes tipos de anotación no son excluyentes entre sí y presentarán mayores ventajas y/o desventajas dependiendo de la naturaleza del sistema de anotación semántica donde se usen.

En cuanto a los métodos de marcado, se ha visto en primer lugar el problema que aparece con el marcado de fragmentos de texto en XHTML/HTML. A continuación se ha estudiado las distintas tecnologías que se usan actualmente para acometer esta labor, destacando *DOM Range* y *XPointer* como recomendaciones de la W3C para el marcado.

Finalmente, se han estudiado de forma resumida las principales características que ofrecen las herramientas de anotación semántica más destacadas y que perduran hasta nuestros días desde que nació el concepto de Web Semántica y se iniciaron las labores para su consecución.

Capítulo 5

FLERSA: Definición de un Sistema Semántico de Gestión de Contenido Web

El futuro tiene muchos nombres. Para los débiles es lo inalcanzable. Para los temerosos, lo desconocido. Para los valientes, la oportunidad.

Victor Hugo

Resumen: Dada la difusión que los CMS están tomando en nuestros días y siendo conscientes de que en un futuro muy cercano van a ser responsables de alojar gran parte del contenido que forma la Web, proponemos enriquecerlos con las tecnologías de la Web Semántica para obtener su variante semántica: Los Sistemas Semánticos de Gestión de Contenidos o también conocidos de forma abreviada como S-CMS. Se trata de sistemas donde los contenidos dirigidos a personas confluyen con metadatos que contienen información semántica sobre dichos contenidos y que son procesables por máquinas. El trabajo realizado en esta dirección ha dado como fruto un S-CMS denominado FLERSA; en este capítulo se estudia cómo se ha diseñado FLERSA, así como las nuevas propuestas, técnicas y fundamentos utilizados en su desarrollo.

5.1. Introducción

En la actualidad, las tecnologías de la Web Semántica han alcanzado el desarrollo y la madurez suficiente que permite pasar de la fase de experimentación a otra de explotación de las mismas y exploración de las posibilidades que ofrecen. Por ejemplo, existen iniciativas como *Linked Open Data* cuyo objetivo es extender la Web con datos compartidos mediante la publicación de varias fuentes de datos expresadas en RDF entre las que se establecen vínculos con diferentes fuentes de datos (Heath y Bizer, 2011).

Nos encontramos ante un nuevo escenario en el que se pronostica un auge de la Web semántica, apoyada por comunidades de investigadores, desarrolladores y empresas cuyo trabajo es coordinado por la W3C, en el que cada vez se alcanzan mayores niveles de abstracción en la representación de información semántica propiciada por el desarrollo en los estándares y especificaciones de representación de datos (XML, RDF, OWL,SKOS y SPARQL).

Estos cambios que se están produciendo conllevan una modificación en la filosofía de gestión de la información por parte de instituciones y organizaciones. Se necesita un nuevo planteamiento y diseño de los servicios y productos de información, así como un análisis de las herramientas a utilizar para ello.

Por otro lado, hay que considerar que la mayor parte de los contenidos de la Web no se encuentran en los formatos de la Web Semántica y que los CMS ocupan una posición importante dentro del ámbito de las tecnologías de contenidos web debido a su amplia difusión y al volumen de información que éstos manejan.

El secreto del éxito que están teniendo este tipo de sistemas es el hecho de que han superado el modelo tradicional de edición centrada en la gestión de ficheros y han evolucionado hacia otro modelo centrado en la estructuración de contenidos y gestión de procesos de trabajo. Características tales como la facilidad de instalación, facilidad de administración y de uso, versatilidad y extensibilidad hacen que exista una gran comunidad de empresas e instituciones que se benefician de las facilidades que proporcionan este tipo de gestores de contenidos para ofrecer servicios y publicar sus contenidos en la Web.

El paradigma de los CMS se fundamenta en la definición de tipos de contenidos, creación de menús de navegación y vinculación de éstos con los contenidos, uso de plantillas para la personalización de la presentación y en una reutilización general de la información. El elemento clave de este tipo de sistemas es que almacenan los contenidos separando estructura, información y presentación.

Por tanto, pensamos que los CMS son un área prometedora donde, aprovechando que disponen de mucha información almacenada, podría ser posible añadir información semántica, procesable por las máquinas, dentro de los contenidos pensados inicialmente para personas. Además, esta misma idea cumple un doble cometido ya que también se podrían enriquecer los contenidos de los CMS usando fuentes externas de datos.

5.2. Objetivos

El principal objetivo es ofrecer las bases para convertir los CMS en sus equivalentes semánticos, los S-CMS, extendiendo así los beneficios que ofrece la Web Semántica. Dichos sistemas deben permitir la realización tanto de anotaciones semánticas manuales como automáticas, así como la realización de búsquedas de información mejoradas basadas en las anotaciones semánticas realizadas.

En el ámbito de los S-CMS existe una brecha inicial, producida por las diferentes necesidades de publicación existentes, por un lado de contenidos y por otro información semántica, que conduce, a su vez, a contemplar de un modo separado la gestión de los CMS, el potencial de metadatos que ofrece a la Web Semántica, y la integración de éstos en aquéllos, y viceversa.

A la vista de las herramientas del apartado 4.4 del capítulo anterior, se observa que ninguna de ellas es apta para conseguir el objetivo principal que se ha establecido, debido a las diferentes deficiencias que presentan: La mayoría son aplicaciones autónomas (standalone) en lugar de web, ninguna está diseñada para ser integrada dentro de un CMS, están centradas en la plataforma en lugar de en el usuario, no todas permiten trabajo colaborativo, para ser usadas requieren de los servicios de un experto o usuario avanzado y no todas permiten la realización de anotaciones semánticas tanto manuales como automáticas.

A la hora de definir un S-CMS, se proponen los siguientes objetivos que éstos deben de cumplir:

- La utilización de estándares abiertos recomendados por la W3C para las tareas relacionadas con el almacenamiento, recuperación y comunicación de información semántica.

- La adopción del entorno web centrado en el usuario como entorno natural de trabajo de las herramientas semánticas que se aporten a los CMS. Además este entorno será colaborativo, en la medida en que lo posibilite el CMS subyacente.

- La generación de anotaciones semánticas de los contenidos, tanto manuales como automatizadas. En particular, es de interés conseguir anotaciones automatizadas ya que el volumen de los contenidos de los CMS puede ser considerable.
- La posibilidad de anotación semántica de los elementos usuales que aparecen en los contenidos de los CMS como pueden ser: Páginas web en formato XHTML, imágenes y elementos multimedia.
- La consistencia de las anotaciones semánticas en los documentos donde se anotan cuando la información es modificada, borrada o movida.
- El almacenamiento de las anotaciones semánticas embebida dentro del documento donde se hacen, así como en la base de conocimiento de un servidor.
- Introducir una mejora sustancial en las tareas de búsqueda y de recuperación de información basada en el uso de los metadatos que generan las anotaciones semánticas.
- La integración total de las herramientas de anotación semántica, búsqueda y recuperación de información dentro de los sistemas CMS.
- La facilidad de uso de las herramientas semánticas que se desarrollen, permitiendo a usuarios estándar utilizar dichas herramientas de manera que se acelere la productividad de información semántica en los sistemas CMS.
- La utilización de ontologías como medio de representación del conocimiento, con la finalidad de facilitar el intercambio de información y posibilitar la comunicación entre sistemas.
- La originalidad del sistema resultante. No se pretende desarrollar desde cero todos los componentes que forman el sistema, es más, sería contraproducente ya que el tiempo y esfuerzo que esto supondría haría que el desarrollo del sistema fuera inviable. Por tanto, la definición del sistema será fruto de la combinación de componentes preexistentes con nuevos componentes, proporcionando los primeros una funcionalidad base a partir de la cual trabajar en los nuevos, reduciendo así el tiempo de desarrollo global del sistema.

La integración de los objetivos anteriores en un CMS dará como resultado el inicio de los CMS Semánticos o S-CMS, en los cuales la publicación de contenidos dirigidos a personas irá acompañada de la publicación de información semántica, en forma de metadatos, procesable por las maquinas, solucionando en gran medida las deficiencias semánticas existentes en el ámbito de los CMS y realizando una nueva aportación que puede ayudar en la extensión de la Web Semántica.

5.3. Requisitos de Diseño

Se establecen siete requisitos de diseño para Sistemas de Anotación Semántica en el artículo de Uren et al. (2006) que a su vez extienden los establecidos por Handschuh et al. (2003a); a continuación se presentan de forma resumida:

- **Requerimiento 1 - formatos estándar:** El uso de estándares puede proporcionar un mecanismo puente que permita que recursos heterogéneos sean accesibles simultáneamente y usuarios y organizaciones cooperen para compartir anotaciones.

- **Requerimiento 2 - diseño colaborativo/centrado en el usuario:** Es importante dotar a los usuarios del conocimiento con interfaces fácil de usar que simplifiquen el proceso de anotación y lo coloquen en el contexto de su trabajo diario, de manera que el entorno en el que los usuarios anoten documentos esté integrado con el de creación, lectura, edición y colaboración. El diseño del sistema también necesita facilitar la colaboración entre usuarios, lo cual es una faceta clave del conocimiento, trabajar con expertos de diferentes campos colaborando y reusando documentos inteligentes.

- **Requerimiento 3 - soporte de ontologías:** Además de soportar formatos de ontología apropiados, el sistema de anotación deberá ser capaz de trabajar con múltiples ontologías. Cada vez que se use una ontología deberá de declararse explícitamente a cual se refiere. Además, el sistema tendrá que hacer frente a los cambios hechos a las ontologías a lo largo del tiempo, tales como incorporación de nuevas clases o modificación de las existentes.

- **Requerimiento 4 - soporte para formatos de documentos heterogéneos:** Los estándares de anotación de la Web Semántica tiene tendencia a asumir que los documentos que se van a anotar están en formatos nativos web tales como HTML y XML. Esta estrategia limita la usabilidad para la gestión del conocimiento. Los documentos estarán en muchos formatos diferentes incluyendo ficheros de procesamiento de textos, hojas de cálculo, ficheros gráficos y mezclas complejas de diferentes formatos.

- **Requerimiento 5 - evolución de documentos:** Este requerimiento trata sobre los aspectos de diseño relativos a qué debe ocurrirles a las anotaciones cuando son revisadas, borradas o movidas de sitio. Los entornos de anotación deben ayudar a los trabajadores del conocimiento a mantener las anotaciones conforme cambian los documentos.

- **Requerimiento 6 - almacenamiento de anotaciones:** El modelo de la Web Semántica asume que las anotaciones serán almacenadas separadamente del documento original. El desacople de contenido y semántica funciona particularmente bien en entornos web donde los autores de las anotaciones no tienen necesariamente algún control sobre los documentos que están anotando. Sin embargo, en entornos de gestión del conocimiento, muchos anotadores están más familiarizados con modelo de procesador de textos. Éstos argumentan que, almacenar las anotaciones como parte de esos documentos es preferible y ayuda a mantener las anotaciones consistentes con nuevas versiones del documento. Se consideran los dos enfoques de almacenamiento de anotaciones para la gestión del conocimiento.

- **Requerimiento 7 - automatización:** La integración de tecnologías de extracción de conocimiento en el entorno de anotación es vital a la hora de proporcionar herramientas de ayuda para el marcado automático en colecciones de documentos grandes.

Además de los requisitos anteriores, a la hora de diseñar el sistema, y considerando que se tratara de un sistema web, se plantearon los siguientes requisitos más específicos:

- **Requisito 8 - integración:** Como disponemos de tiempo finito y recursos limitados, no se trata de desarrollar toda una infraestructura orientada a la obtención una herramienta, sino que se trata de realizar una integración de distintos componentes y del desarrollo de forma modularizada del sistema acorde con la interfaces de programación y/o comunicación que proporcionen los componentes constituyentes del sistema.

- **Requisito 9 - interfaz de usuario web:** El ámbito de trabajo natural de los CMS es el navegador web, así que su variante semántica debe ser capaz de trabajar y administrase desde el mismo entorno.

- **Requisito 10 - compatibilidad multi-navegador:** Puesto que el sistema usará una interfaz de usuario web y hoy día existen gran variedad de navegadores web, cada cuál con sus peculiaridades, se buscará dentro de lo posible la compatibilidad de la herramienta al menos con los navegadores más difundidos.

Analizando los requisitos de diseño generales y específicos podemos observar que aparecen algunos conflictos. El requerimiento 4 propone la creación de un sistema de anotación genérico que de soporte a cualquier tipo de documentos, no sólo a documentos web, mientras que el requisito 9 fija que el ámbito de

trabajo del sistema que se está definiendo al entorno web. El requisito 9 es más específico que el 4 y por lo tanto relajamos las condiciones que se establecen en él y nos centramos en el entorno web.

5.4. Arquitectura

Con objeto de satisfacer los requisitos enunciados en el apartado 5.3 anterior, y teniendo en cuenta el trabajo que Le y Lau (2006) ha realizado en éste ámbito, se define la siguiente arquitectura de sistema que consta de cuatro niveles: *El nivel de gestión de información, el nivel núcleo, el nivel semántico del servidor y el nivel web*. Véase a continuación la figura 5.1.

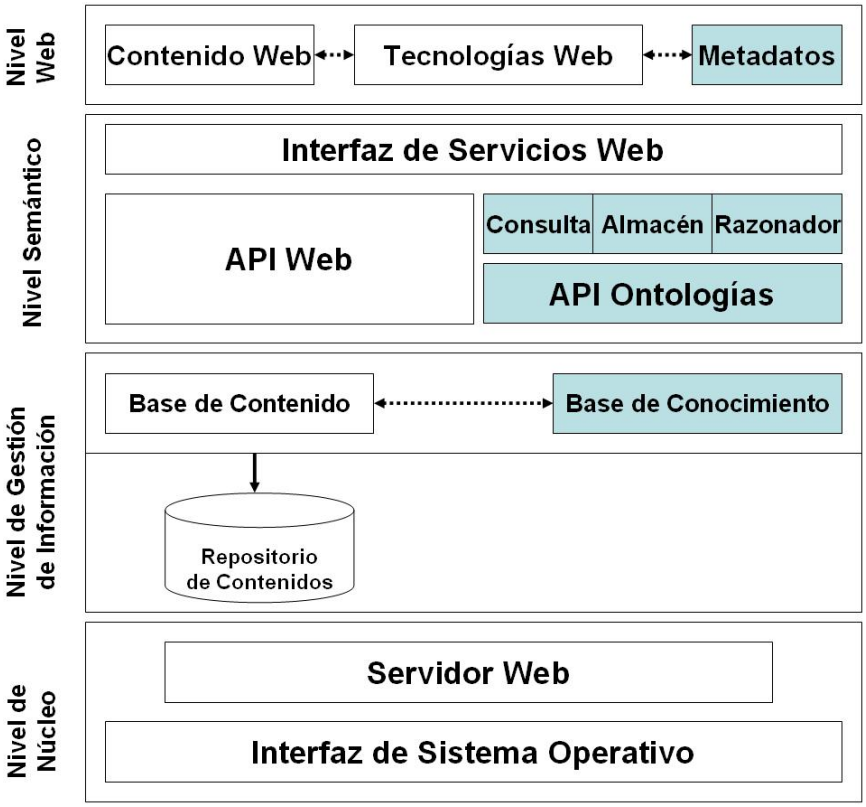


Figura 5.1: Arquitectura del sistema FLERSA.

A **nivel de núcleo** nos encontramos con el sistema operativo, los servicios de red y el servidor web. Se trata del nivel que aporta la infraestructura software y de comunicaciones necesaria para instalar el resto de componentes software que componen el sistema. En particular, se necesita un sistema operativo multitarea con capacidad para trabajar con la familia de protoco-

los de red TCP/IP y un servidor web que permita el alojamiento de páginas web dinámicas.

En el **nivel de gestión de información** está formado por un componente del sistema llamado *repositorio de contenidos* encargado del almacenamiento tanto de contenido web (*base de contenido*), como de información semántica sobre dicho contenido. También es el nivel donde se almacena la *base de conocimiento* compuesta por las diversas ontologías con las que trabaja sistema. Por tanto, se va a necesitar un gestor de bases de datos para el alojamiento de la información del repositorio de contenidos; en cuanto a la información semántica y a la base de conocimiento, se evaluará la posibilidad de utilizar este mismo gestor de bases de datos u otra herramienta de almacenamiento especializada.

En el **nivel semántico** de servidor es donde se desarrollan los servicios de aplicación. Todo el tráfico de mensajes entre los clientes web que demandan información y los servicios que los proporcionan tienen lugar en este nivel a través de la *interfaz de servicios web*. En este nivel confluyen los servicios de contenidos, encargados de trabajar con información dirigida a personas, con los servicios semánticos, encargados de enriquecer esta información con metadatos dirigidos a las computadoras. Para el desarrollo de los servicios de contenidos se utilizan las primitivas que ofrecen librerías de programación web o *APIs web*. Para el desarrollo de los servicios semánticos se utilizan librerías semánticas o *APIs para trabajar con ontologías*. En general, entre las funciones más usadas que ofrecen estas APIs cabe destacar las facilidades de almacenamiento y recuperación de información, las facilidades para trabajar con objetos visuales en la programación de la interfaz de usuario y las facilidades para trabajar con ontologías e información semántica.

Por último contamos con un **nivel de interfaz web**, situado en el nivel superior de abstracción de la arquitectura del sistema, desde donde el usuario realiza toda la interacción con el sistema semántico o S-CMS. En este nivel conviven los contenidos de los documentos web, los metadatos vinculados a éstos y las tecnologías web encargadas de modificar en tiempo de ejecución los documentos web para dotarlos de anotaciones semánticas en forma de metadatos, así como también de realizar la gestión de mensajes oportuna, mediante el uso de servicios del lado servidor, para aportar la funcionalidad de la herramienta.

5.5. Requerimientos para la Anotación Semántica

En esta sección proponemos una serie de requisitos que debería cumplir cualquier sistema de anotación semántica. Los requisitos se pueden resumir en tres, y son:

1. El uso de ontologías tanto a nivel de infraestructura durante el proceso de creación de anotaciones semánticas, como a nivel de referencia durante el proceso de asociación de significado a los textos marcados.

2. El uso de algún esquema de anotación propuesto por la W3C, como por ejemplo Annotea.

3. El uso de estándares de la W3C para el marcado semántico tales como RDF, RDFa u OWL.

A continuación, en las distintas subsecciones, se procede al análisis de cada uno los requisitos propuestos y a la justificación de las ventajas que suponen su satisfacción para cualquier sistema de anotación semántica.

5.5.1. Uso de Ontologías

Según se estudió en el apartado 3.3.6, la definición más difundida de **ontología** es la ofrecida por Gruber (1993) y ampliada por Studer et al. (1998), que establece que una ontología es "*una especificación explícita y formal de una conceptualización*". Las implicaciones de esta definición en el ámbito de la Web Semántica son las siguientes:

- **Formal**: Se refiere al hecho de que una ontología debe ser comprensible para las máquinas.
- **Explícita**: Significa que el tipo de conceptos usados y las restricciones en ellos son definidos explícitamente.
- **Compartida**: Refleja la noción de que la ontología captura conocimiento consensuado, es decir, no es particular de un individuo, sino aceptado por un grupo.
- **Conceptualización**: Se refiere al modelo abstracto de algún fenómeno del mundo del que se identifican conceptos relevantes para ese fenómeno.

En el apartado 3.3.6 del capítulo 3 se realizó un estudio de los elementos de una ontología y tipos de ontología que existen según su expresividad o nivel de abstracción. En el presente apartado se van a justificar las ventajas que presenta el uso de ontologías a la hora de realizar anotaciones semánticas.

La función más importante de las ontologías es almacenar conocimiento de forma que pueda utilizarse por sistemas automáticos capaces de realizar deducciones a partir de la variedad de relaciones entre conceptos.

La posibilidad de compartir la estructura del conocimiento representado es uno de los principales objetivos en el desarrollo de ontologías. Por ejemplo, supongamos el caso de varios sitios web que contienen información médica o proporcionan servicios de comercio electrónico farmacéutico. Si todos estos sitios utilizan y publican la misma ontología sobre los términos que manejan, un agente software podrá consultar y agregar información para proporcionar un servicio más completo del que se ofrece por separado. A su vez, este conocimiento podrá utilizarse como entrada en otras aplicaciones o como lenguaje intermedio para intercambio de datos.

La reutilización es otro de los valores fundamentes de las ontologías, ya que permite ahorrar esfuerzo en la adquisición y codificación del conocimiento. Así, para construir una ontología se pueden integrar varios modelos previos que describan partes menores del dominio. Y al contrario, también es posible tomar una ontología más general y especializarla para describir la parte del dominio que más nos interese.

La representación del conocimiento mediante declaraciones explícitas sobre el dominio garantiza que, en caso de que cambie la información de que se dispone, se puedan modificar estas afirmaciones sin que sea necesario reescribir el software que utiliza la ontología. De esta forma se consigue separar el conocimiento declarativo del procedimental.

Una vez construida la ontología dispondremos de una especificación del dominio estudiado. Dado que esta descripción tendrá un carácter formal, se podrá analizar el conocimiento reflejado para validarlo y verificarlo manual y automáticamente, de manera que pase a formar parte de un repositorio de confianza.

A modo de resumen de las argumentaciones anteriores, se propone el uso de ontologías a la hora de realizar anotaciones semánticas debido a las múltiples ventajas que éstas ofrecen, entre las que cabe destacar:

- Permiten compartir un entendimiento común de la realidad que se está observando, de la estructura de la información (un vocabulario común) entre personas o agentes (software).
- Permiten la reutilización de conocimiento de dominio, la misma conceptualización que se usó en un dominio puede aplicarse a otros.
- Posibilitan la integración de conocimiento, mediante la integración de ontologías existentes.
- Sirven para explicitar suposiciones del dominio, ya que todo lo que no está dicho no existe.
- Separan el conocimiento del dominio del conocimiento operacional.

- Permiten realizar tareas de análisis de dominios de conocimiento y extracción de nueva información gracias a la capacidad de inferencia que poseen. Disponen de asertos, reglas y axiomas equivalentes a la Lógica de Descripciones.

A la hora de realizar anotaciones semánticas pueden ser especialmente útiles dos tipos: Ontologías de dominio y de aplicación. Las **ontologías de dominio** pueden aportan una taxonomía de conceptos que proporcionan un vocabulario consensuado para definir interrelaciones entre los marcados en páginas web y la semántica de la información que contienen (conceptos o categorías de los que tratan). Las **ontologías de aplicación** son ideales para la elaboración de una estructura base de anotación semántica. Las anotaciones semánticas, desde un punto de vista ontológico, pueden ser consideradas como elementos o individuos que representan anotaciones concretas en documentos siguiendo el modelo formal definido en la ontología base. Estos individuos corresponden a hechos concretos del concepto Anotación que se define en la ontología base.

5.5.2. Annotea - Esquema de Anotación Semántica

Annotea (Kahan y Koivunen, 2001) es un proyecto de la W3C que especifica la estructura de anotación para documentos web, haciendo énfasis en el uso colaborativo de anotaciones. Permite realizar anotaciones en las páginas web sin que el documento original sufra ninguna transformación.

El formato principal de Annotea es RDF y el tipo de documentos que es capaz de anotar está limitado a HTML o XML. Utiliza tecnología XPointer, estudiada en la sección 4.3.4, para la localización de las anotaciones dentro de un documento.

El enfoque de Annotea se centra en un estilo semi-formal de anotación, donde las anotaciones son sentencias de texto libre sobre documentos. Estas sentencias deben tener metadatos y pueden ser clasificadas de acuerdo con un esquema RDF de complejidad arbitraria definido por el usuario.

El esquema básico de anotación de Annotea esta compuesto por una clase RDF llamada *"Annotation"* que dispone de las siguientes propiedades:

- **Annotates:** Asocia una anotación con el recurso con el que se anota.
- **Author:** El nombre de la persona u organización responsable de la creación de la anotación.
- **Body:** Asocia el contenido de la anotación con el recurso de la anotación.

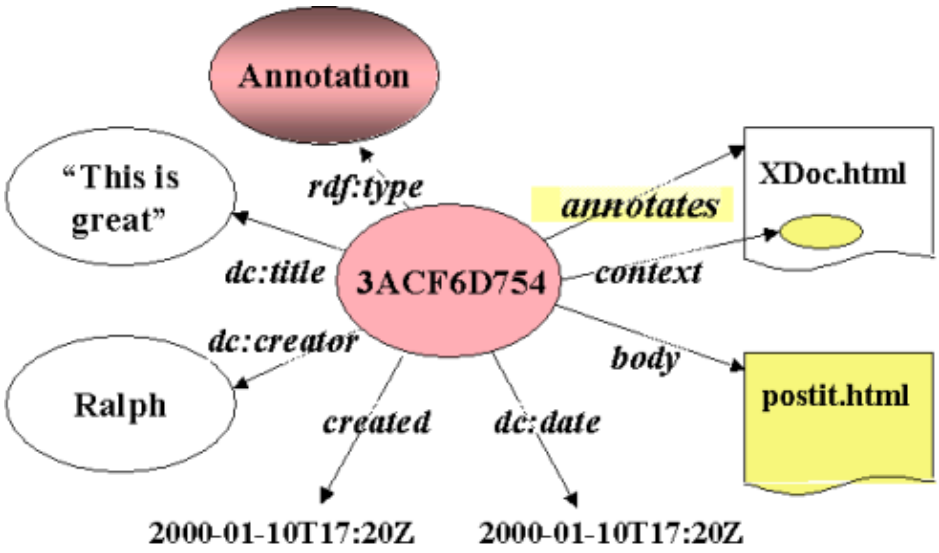


Figura 5.2: Grafo RDF de una anotación con estructura Annotea.

- **Context:** El contexto dentro del recurso especificado en la propiedad *"Annotates"*.
- **Created:** Fecha y hora de creación de la anotación.
- **Modified:** Fecha y hora de última modificación de la anotación.
- **Related:** Una relación entre la anotación y recursos adicionales menos específicos que los especificados en la propiedad *"Body"*.

Además de las anteriores propiedades, también se dispone de la propiedad *"Type"* del vocabulario RDF que permite referenciar el tipo de anotación que se está realizando dentro de los siguientes tipos posibles definidos: *Comment, seealso, question, explanation, example, comment, change y advice.*

5.5.3. Uso de Estándares W3C para el Marcado Web

En este apartado se va a ilustrar cómo usar Annotea junto a los lenguajes de marcado propuestos por la W3C, tales como RDF y RDFa, para realizar anotaciones semánticas en documentos web.

Tomando el framework Annotea como estructura de anotación base, la traducción de la anotación que aparece en la figura 5.2 a lenguaje RDF sería la que aparece en el ejemplo 4.10.

```
<div  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:a="http://www.w3.org/2000/10/annotation-ns#"
          xmlns:t="http://www.w3.org/2000/10/annotationType#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      about="http://ex.com/XDoc.html#3ACF6D754">

    <span typeof="a:Annotation"/>
    <span resource="file:///home/swick/.amaya/annotations/postit.
       html" rel="a:body" />
    <span resource="http://ex.com/XDoc.html" rel="a:annotates"/>
    <span resource="http://ex.com/XDoc.html#xpointer(id('Main')/p
       [2])" rel="a:context"/>
    <span property="a:created">2000-01-10T17:20Z</span>
    <span property="dc:title">This is great</span>
    <span property="dc:creator">Ralph Swick</span>
    <span property="dc:date">2000-01-10T17:20Z</span>
 </div>
```

Ejemplo 5.1: Anotación RDFa según la estructura Annotea.

Se puede apreciar, en el ejemplo 5.1, que en las cuatro primeras líneas se procede a la definición de los espacios de nombres: En primer lugar el de RDF, a continuación el framework Annotea y por ultimo el vocabulario Dublín Core. Después se pasa a describir el recurso, de tipo *"Annotation"*. Se está anotando la página *"XDoc.html"*; el contexto, usando la tecnología XPointer, es el segundo párrafo del contenedor HTML con identificador *"Main"*. Por último se detallan el creador de la anotación, las fechas de creación y el cuerpo de la anotación definido con la etiqueta *"body"*, que hace referencia a un fichero independiente perteneciente a la herramienta de anotación Amaya.

Los clientes de Annotea crean anotaciones como las del ejemplo 5.1 y las envían mediante protocolo HTTP al lado servidor donde son almacenadas en una base de datos específica RDF; se trata por tanto de un proceso de anotación externo en el cual el documento original permanece inalterado y las anotaciones se almacenan en un servidor dedicado.

Al igual que en el ejemplo anterior, tomando el framework Annotea como estructura de anotación base, la traducción de la anotación que aparece en la figura 5.2 a lenguaje RDF quedaría como aparece en el ejemplo 5.2.

Se puede apreciar en el ejemplo 5.2, que en las cuatro primeras líneas se procede a la definición de los espacios de nombres: En primer lugar el de RDF, a continuación el framework Annotea y por ultimo el vocabulario Dublín Core. En la línea quinta se indica el recurso sobre el que se va a realizar la descripción. En el resto de líneas, al igual que en el ejemplo 5.1, se describen el resto de campos de la estructura de Annotea y otros del vocabulario Dublín Core.

```
<r:RDF xmlns:r="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:a="http://www.w3.org/2000/10/annotation-ns#"
       xmlns:t="http://www.w3.org/2000/10/ annotationType#"
       xmlns:d="http://purl.org/dc/elements/1.1/">

 <r:Description about="http://ex.org/Annotation/3ACF6D754">
   <r:type resource="a:Annotation"/>
   <a:annotates r:resource="http://ex.com/some/XDoc.html"/>
   <a:context r:resource='http://ex.com/XDoc.html#xpointer(id("Main
       ")/p[2])'/>
   <d:creator>Ralph Swick</d:creator>
   <d:title>This is great</d:title>
   <a:created>2000-10-10T17:20Z</a:created>
   <d:date>2000-01-10T17:20Z</d:date>
   <a:body r:resource="file:///home/swick/.amaya/annotations/postit
       .html"/>
 </r:Description>

</r:RDF>
```

Ejemplo 5.2: Anotación RDF basada en estructura Annotea.

Cabe destacar que el ejemplo 5.1 corresponde a una anotación incrustada en el documento que se anota cuyo principal objetivo es no repetir contenido cuando se están anotando datos estructurados, de forma que el código XHTML resultante es totalmente válido. En cambio, el ejemplo 5.2 no puede ser incrustado dentro del documento que se anota, necesita ser almacenado en un fichero o también podría ser generado dinámicamente, pero en cualquier caso se trataría de una anotación semántica externa.

La anotación RDFa del ejemplo 5.1 es totalmente equivalente a la del ejemplo 5.2. Para obtener su traducción a RDF/XML tan sólo es necesario pasarle un procesador de RDFa, como por ejemplo RDFa Distiller[1]. A continuación, en el ejemplo 5.3, se muestra la salida que produciría si aplicamos el procesador RDFa Distiller a la anotación RDFa del ejemplo 5.1.

El ejemplo 5.3 también es una anotación semántica en RDF usando la infraestructura Annotea. Sintácticamente es diferente y está mejor estructurado que el ejemplo 5.2, aunque mantiene la misma expresividad e igual contenido semántico que éste, por lo que son equivalentes.

[1] http://www.w3.org/2007/08/pyRdfa

```
<r:RDF xmlns:r="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:a="http://www.w3.org/2000/10/annotation-ns#"
       xmlns:t="http://www.w3.org/2000/10/ annotationType#"
       xmlns:d="http://purl.org/dc/elements/1.1/">
  <a:Annotation rdf:about="http://ex.com/XDoc.htm#3ACF6D754">
        <a:annotates rdf:resource="http://ex.com/XDoc.htm"/>
        <a:context rdf:resource="http://ex.com/XDoc.htm#xpointer(id
            ('Main')/p[2])"/>
        <a:created>2000-01-10T17:20Z</a:created>
        <a:body rdf:resource="file:///home/swick/.amaya/annotations
            /"/>
        <dc:title>This is </dc:title>
        <dc:date>2000-01-10T17:20Z</dc:date>
        <dc:creator>Ralph Swick</dc:creator>
  </a:Annotation>
</r:RDF>
```

Ejemplo 5.3: Anotación RDF mejorada basada en estructura Annotea.

5.6. Ontologías en FLERSA

Desde que se definieron los objetivos del sistema, al principio del capítulo, el uso de ontologías ha estado siempre presente debido a las múltiples ventajas que éstas ofrecen.

En este apartado se presenta "FLERSA-Ontology", la ontología que se ha diseñado expresamente para que realice la función de infraestructura para la anotación semántica de sistemas tipo S-CMS.

5.6.1. Infraestructura de Anotación Semántica

Las anotaciones semánticas, desde un punto de vista ontológico, son consideradas elementos o individuos que representan anotaciones concretas en documentos siguiendo el modelo formal definido en la ontología base. Estos individuos corresponden a hechos concretos del concepto "anotación" (Annotation) que se define en la ontología base.

Una vez tomada la decisión de usar ontologías, se consideran las alternativas disponibles en el contexto de tecnologías de la Web Semántica. Finalmente se opta por adoptar el framework Annotea como modelo de referencia a seguir ya que fue diseñado por la W3C específicamente como estructura de anotación para documentos web y satisface el requerimiento 1 del apartado 5.3.

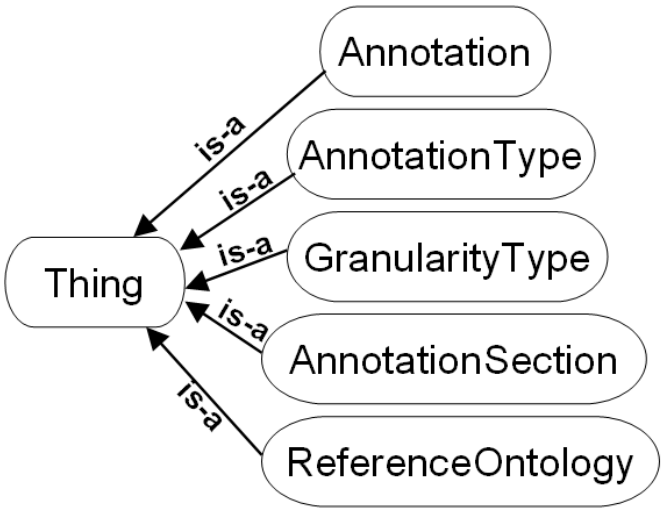


Figura 5.3: Taxonomía de clases de infraestructura.

Puesto que la Web es el ámbito de trabajo para el que se necesita la ontología base de anotación, se ha reutilizado, adaptado y enriquecido la estructura por defecto definida en Annotea. De aquí en adelante nos referiremos a la ontología base para la anotación semántica con el nombre de FLERSA-Ontology.

El esquema básico de FLERSA-Ontology está compuesto por cinco clases OWL, como se puede observar en la figura 5.3.

La clase principal es *"Annotation"* y dispone de las mismas propiedades con las que contaba la estructura de anotación que define Annotea. En la figura 5.4 se muestra la estructura de la clase.

Las propiedades de las que dispone son las siguientes:

- **Annotates**: Al igual que en Annotea, asocia una anotación con la página web (el recurso) con el que se anota.
- **Author**: El nombre del usuario responsable de la creación de la anotación. En el entorno de los CMS suele corresponder a un nombre de usuario representada en forma de cadena de caracteres.
- **Body**: Fragmento de texto de la página Web objeto de la anotación semántica. Se almacena el texto completo para facilitar futuras labores de búsqueda. Este campo se omite en el caso de que el objeto que se anote sea multimedia.

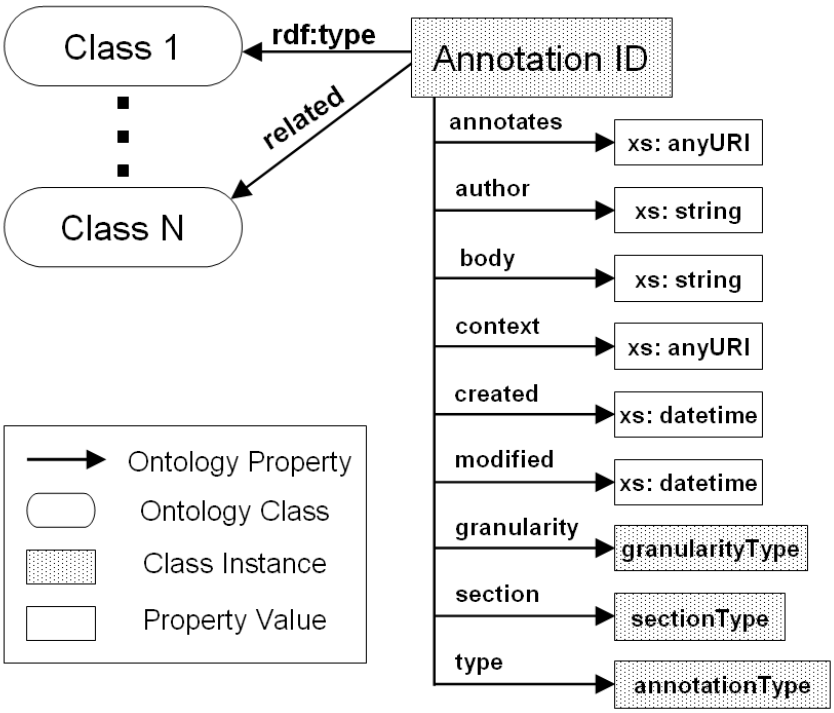


Figura 5.4: Instancia de la clase *"Annotation"*.

- **Context**: El contexto dentro de la página Web especificado en la propiedad `Annotates`. Corresponde con la URI que delimita la posición donde se encuentra el texto u objeto multimedia que se anota. Ej. `http://w3.ex.org/p.htm#6543`
- **Created**: Al igual que en Annotea, sirve para indicar la fecha y hora de creación de la anotación.
- **Modified**: Ídem con la fecha y hora de ultima modificación de la anotación.
- **Related**: Establece una relación entre la anotación y las ontologías de referencia que se usan a modo de taxonomías. Sirve para asociar una anotación con el concepto del que se habla en ella. Es una de las propiedades más importantes puesto que en ella se especifica el concepto del que trata la anotación, y en el futuro va a ser muy útil para hacer búsquedas sobre la base de datos de anotaciones.

Las propiedades que han aparecido anteriormente, se puede decir que se han *heredado* del framework Annotea. Las siguientes propiedades han sido concebidas específicamente para FLERSA-Ontology:

- **Granularity**: Asocia a la anotación un concepto dentro de la taxonomía que ofrece la clase *"GranularityType"* indicando el tipo de granularidad de la anotación (véase figura 5.5). Los tipos de granularidad son: *Carácter, palabra, frase, párrafo o texto libre*. El tipo de granularidad se debe asignar de forma automática dependiendo de las características del fragmento de texto que se anote. Esta propiedad no se usa a la hora de anotar objetos multimedia.

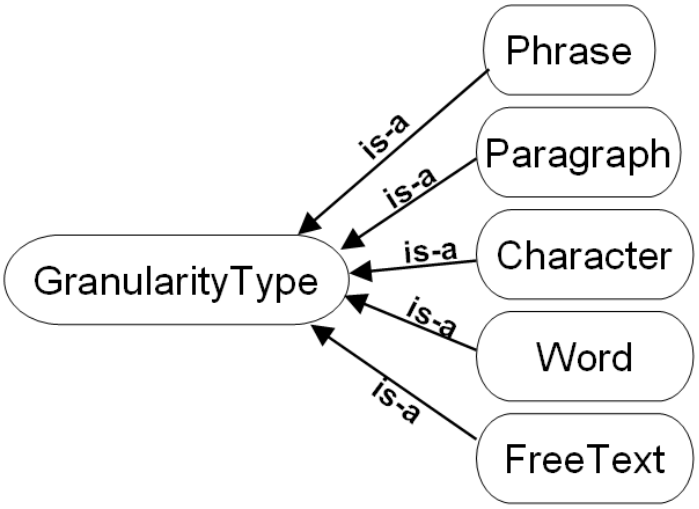


Figura 5.5: Taxonomía de tipos de granularidad.

- **Section**: Asocia a la anotación un concepto dentro de la taxonomía que ofrece la clase *"AnnotationSection"* indicando la sección dentro de la página Web donde se ha realizado de la anotación (véase figura 5.6). Los tipos de secciones principales son: *Texto e imagen*.

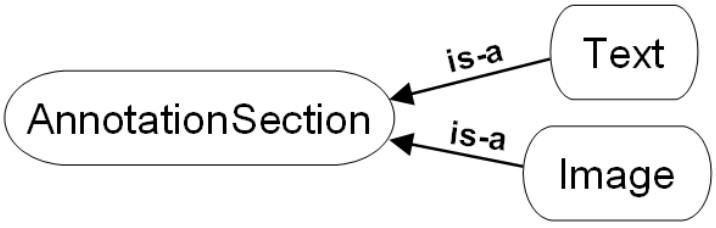


Figura 5.6: Taxonomía de tipos de sección.

- **Type**: Asocia a la anotación un concepto dentro de la taxonomía que ofrece la clase *"AnnotationType"* indicando el tipo de anotación que se

ha realizado (véase figura 5.7). Los tipos de anotaciones que se pueden realizar son: *example, advice, change, seealso, explanation, question y comment*. Estos tipos se han heredado del framework Annotea.

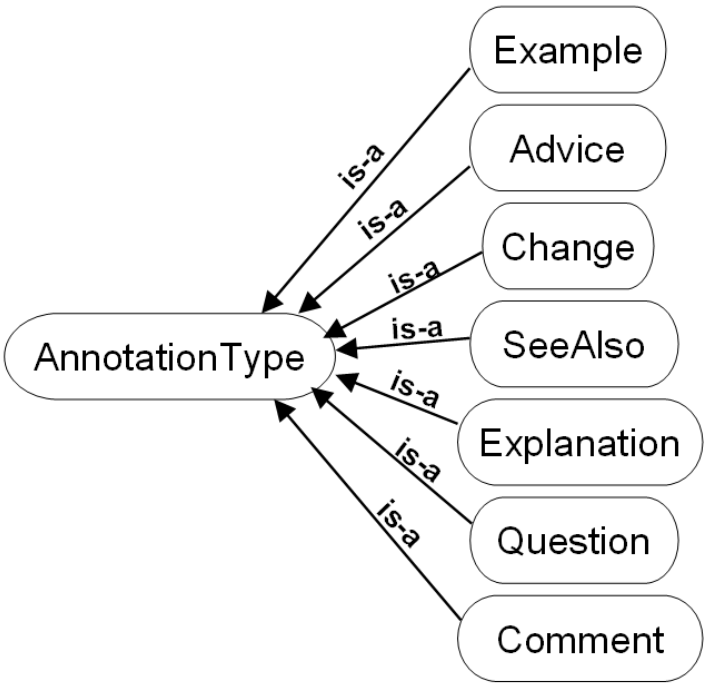


Figura 5.7: Taxonomía de tipos de anotación.

Las clases *"AnnnotationType"*, *"GranularityType"* y *"AnnotationSection"* no proceden del framework Annotea, son clases totalmente nuevas que se han agregado para modelar las taxonomías de conceptos con las que se vincularán las propiedades de instancia *granularity, section y type* explicadas anteriormente.

En la figura 5.8, se puede observar un ejemplo de anotación semántica. Se trata de un individuo o instancia del concepto *"Annotation"* que representa una anotación sobre un documento web con las propiedades que se indican en el grafo RDF.

Para concluir este apartado, en el listado 5.4 se presenta la traducción del grafo de la figura 5.8 a lenguaje RDF. Se puede observar cómo una anotación semántica se traduce en una serie de sentencias RDF. La finalidad es almacenar todas estas sentencias RDF para constituir una Base de Conocimiento y usar toda esta información para realizar búsquedas "inteligentes" e inferir nueva información en base a las ontologías y anotaciones realizadas.

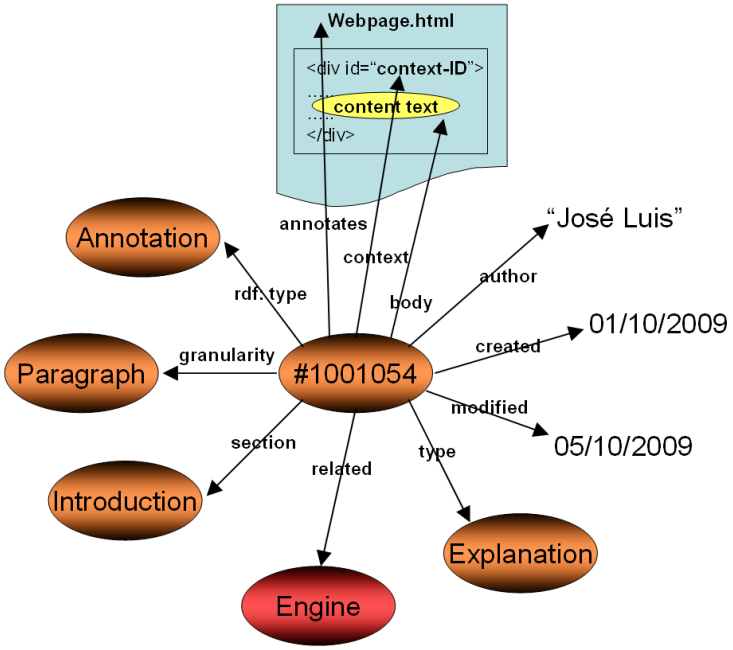


Figura 5.8: Ejemplo de anotación con FLERSA-Ontology.

```
<rdf:RDF xmlns:a="http://www.scms.es/annotation#"
         xmlns:auto="http://www.scms.es/automobile#"
         xmlns:dc="http://purl.org/dc/elements/1.1/">
   <rdf:Description rdf:about="http://site/webpage.htm">
      <dc:title>Annotation example</dc:title>
      <dc:date>2009-02-05 19:06:22</dc:date>
   </rdf:Description>
   <a:Annotation rdf:about="http://site/webpage.htm#1001054">
      <a:Annotates rdf:resource="http://site/webpage.htm"/>
      <a:Context rdf:resource="http://site/webpage.htm#1001054
          "/>
      <a:Body>is a machine designed to convert energy into
          useful mechanical motion</a:Body>
      <a:Author>José Luis Navarro</a:Author>
      <a:Granularity rdf:resource="a:FreeText"/>
      <a:Related rdf:resource="auto:Engine"/>
      <a:Section rdf:resource="a:Introduction"/>
      <a:Created >2009-10-01 19:44:45</a:Created>
      <a:Modified >2009-10-05 19:44:45</a:Modified>
   </a:Annotation>
</rdf:RDF>
```

Ejemplo 5.4: Anotación RDF según la estructura Annotea.

5.6.2. Gestión de Ontologías

El término gestión de ontologías se refiere al conjunto de tareas necesarias para poder utilizar ontologías distintas de la ontología base de anotación (FLERSA-Ontology) con el objetivo de realizar anotaciones semánticas adicionales.

Las ventajas que se obtienen posibilitando la utilización de múltiples ontologías durante las tareas de anotación semántica son muchas. A continuación se explican las más destacadas:

- Posibilidad de asociar conceptos de taxonomías de distintos ámbitos a las anotaciones semánticas de forma sencilla. Esta característica es muy útil, por ejemplo para describir de manera formal los conceptos de los que se hablan en una anotación semántica. En la línea número 14 del ejemplo 5.4, se puede observar como se usa la propiedad *"Related"* para asociar a la anotación el concepto *"auto:engine"* con objeto de indicar que el cuerpo de la anotación, propiedad *"body"* en la línea 11, trata sobre el concepto "motor".

- Posibilidad de enriquecer el vocabulario del sistema de anotación semántica. Gracias a la capacidad de manejar múltiples ontologías, podemos usar cualquier microformato en una anotación semántica mediante el uso de la ontología que modele dicho microformato. Así, por ejemplo, si se está interesado en usar un microformato en las anotaciones tipo RSS, ATOM, FOAF, Dublín Core o incluso RDFS, bastaría con incorporar el modelo de dicho microformato a la Base de Conocimiento para adquirir la nueva funcionalidad. En las líneas número 5 y 6 del ejemplo 5.4, se puede observar cómo se usan propiedades del vocabulario Dublin Core para describir aspectos del documento donde se realiza la anotación semántica.

- Posibilidad para crear individuos de conceptos con la particularidad de que sus propiedades toman valores de las anotaciones semánticas realizas en un documento. Ésta es la más importante de todas las ventajas y, expresado con otras palabras, permite usar las anotaciones semánticas de un documento como valores a partir de las cuales definir nuevas instancias de una clase. Esta característica se fundamenta en el uso de RDF-Schema para definir el tipo de concepto del que se está hablando en una página; también es importante la posibilidad de usar las anotaciones semánticas realizadas en una página web para establecer los valores del concepto que se defina. A continuación, en la línea 4 del ejemplo 5.5, se puede observar cómo se utiliza el vocabulario FOAF para definir una instancia de la clase *"Person"*. Además, como se puede observar en las líneas 5 y 6, se define el valor que toman las

propiedades *"name"* y *"phone"* vinculándolos a los mismos recursos de los que hablan las dos anotaciones que aparecen en las líneas 8 y 16.

```
1  <rdf:RDF xmlns:a="http://www.scms.es/annotation#"
2           xmlns:dc="http://purl.org/dc/elements/1.1/"
3           xmlns:foaf="http://xmlns.com/foaf/0.1/">
4     <foaf:Person rdf:about="http://site/webpage.htm">
5        <foaf:name rdf:resource="http://site/webpage.htm#1001054
             "/>
6        <foaf:phone rdf:resource="http://site/webpage.htm
             #1001055"/>
7     </foaf:Person>
8     <a:Annotation rdf:about="http://site/webpage.htm#1001054">
9        <a:annotates rdf:resource="http://site/webpage.htm"/>
10       <a:context rdf:resource="http://site/webpage.htm#1001054
             "/>
11       <a:body>Michael Kenneth Mann</a:body>
12       <a:author>José Luis Navarro</a:author>
13       <a:granularity rdf:resource="a:FreeText"/>
14       <a:created >2011-11-01 20:01:40</a:created>
15    </a:Annotation>
16    <a:Annotation rdf:about="http://site/webpage.htm#1001055">
17       <a:annotates rdf:resource="http://site/webpage.htm"/>
18       <a:context rdf:resource="http://site/webpage.htm#1001055
             "/>
19       <a:body>1 800 247 72 46</a:body>
20       <a:author>José Luis Navarro</a:author>
21       <a:granularity rdf:resource="a:FreeText"/>
22       <a:created >2011-11-01 20:04:15</a:created>
23    </a:Annotation>
24 </rdf:RDF>
```

Ejemplo 5.5: Anotación RDF según la estructura Annotea.

Si volvemos a la figura 5.3, cabe comentar que la clase *"ReferenceOntology"* es la que se encarga de gestionar qué ontologías participan. Se trata de un metaconcepto que únicamente interviene a la hora de gestionar la interfaz de usuario del Sistema de Anotación Semántica. Dispone de cuatro propiedades:

- **modelURI**: Aquí se especifica la URI donde se accede al modelo.
- **prefix**: Valor de prefijo que se asigna al modelo para referenciarlo usando namespaces.
- **isTaxonomy**: Valor booleano que indica si el modelo se usa como taxonomía de referencia en la definición de anotaciones semánticas.

- **isVocabulary**: Valor booleano que indica si el modelo se usa como vocabulario a partir del cual definir anotaciones semánticas.

A la hora de definir un Sistema de Anotación Semántico que soporte ontologías desde la perspectiva estudiada en este apartado, es necesario incorporar un módulo de administración, accesible por los usuarios administradores del sistema, que permita configurar varios los aspectos estudiados anteriormente:

- Qué ontologías se importan en la Base de Conocimiento.
- Qué ontologías se usan como taxonomías donde buscar conceptos de referencia.
- Qué ontologías se usan como nuevos vocabularios con los que realizar anotaciones semánticas.

5.7. Técnica de Definición de Rangos Flexibles de Texto

La *técnica de definición de rangos flexibles* fue presentada en el artículo de Navarro-Galindo y Samos (2010a). En esta sección se realiza una revisión de la misma, incluyendo ejemplos y describiendo cómo tiene lugar el proceso de marcado. Comienza con el estudio de las condiciones que originaron la creación de la técnica. Después se explica la técnica de delimitación e identificación de fragmentos para textos simples y posteriormente para otros más complejos. La sección concluye con una descripción del proceso de creación de rangos flexibles a modo de resumen.

5.7.1. Requisitos

En el artículo de Uren et al. (2006) mencionado en el apartado 5.3 del presente capítulo se realiza un estudio de los requisitos que debe satisfacer un Sistema de Anotación para Gestión del Conocimiento. En los requisitos 5 y 6 habla de aspectos relacionados con los métodos de marcado de texto, sin entrar en detalles técnicos. El requerimiento 5 trata sobre la evolución de documentos, especificando que se debe dar solución a los aspectos de diseño relativos a qué les ocurre a las anotaciones cuando son revisadas, borradas o movidas de sitio. Los entornos de anotación necesitan ayudar a los ingenieros del conocimiento para que mantengan anotaciones apropiadas conforme los documentos cambian. El requisito 6 trata sobre el almacenamiento de anotaciones, especificando que en entornos donde el usuario tenga control sobre los

documentos que anota, es preferible almacenar las anotaciones incrustadas en el propio documento, ya que ayuda a mantener la consistencia del mismo.

La tecnología XPointer es una recomendación de la W3C para identificar fragmentos dentro de recursos URI; es la tecnología que usa el framework Annotea para especificar la parte del documento sobre la que se realiza la anotación. XPointer es una tecnología robusta como método de localización del componente de texto al que se refiere una anotación, pero presenta problemas cuando se realizan modificaciones sobre un documento sobre el que existen anotaciones. Normalmente, cuando editamos un documento solemos añadir, borrar y/o alterar el orden de los párrafos que lo componen, lo que provoca que se desajusten los puntos de anclaje definidos en XPointer para las anotaciones, por lo que sería necesario repetir el proceso de anotación cada vez que se modifica sustancialmente un documento.

La técnica de definición de rangos flexibles para documentos web es una alternativa a la tecnología XPointer basada en el estándar RDFa. Su principal objetivo es permitir que las anotaciones semánticas definidas siguiendo esta técnica soporten la evolución del documento web donde se encuentren de forma más efectiva que otras técnicas. La técnica también funciona bien cuando se usa para definir anotaciones sobre fragmentos de texto que se solapan. El término *Rangos Flexibles* indica que las anotaciones pueden ser definidas sobre diferentes rangos de texto dentro de un documento web y sobre elementos multimedia.

Estudiado el requisito 5 y vistos los problemas que presenta el uso de la tecnología XPointer, se ha investigado cómo usar el Modelo de Objetos de Documento Rango (DOM Range), ya que explotando la funcionalidad que aporta, permite delimitar e identificar los rangos de texto sobre los que se realizarán anotaciones semánticas. Esta delimitación se realiza haciendo uso de elementos HTML y, por tanto, posibilita alcanzar el objetivo de que la anotación semántica quede almacenada dentro el propio documento.

La técnica usa la especificación *DOM Range* recomendada por el W3C, ya que la funcionalidad que proporciona permite la identificación y delimitación de fragmentos de texto sobre los que se realizarán las anotaciones semánticas. Esta delimitación se lleva a cabo usando elementos HTML y, por lo tanto, se almacena incrustada dentro del mismo documento.

Por otro lado, para satisfacer el requisito 6 contamos con RDFa, un lenguaje de reciente aparición propuesto por la W3C como lenguaje de anotación semántica incrustada para documentos web. Este lenguaje, unido a la capacidad de las tecnologías que implementan la especificación DOM Range para delimitar los rangos de texto que se van a anotar, nos permite realizar anotaciones incrustadas en RDFa dentro del documento web que se anota, salvando los problemas que presenta XPointer.

5.7.2. Delimitación e Identificación Simple

El objetivo es, a partir de la selección de texto que haga el usuario, delimitar de alguna manera el fragmento de texto que se pretende anotar. Para ello, en primer lugar, se necesita identificarlo unívocamente para, posteriormente, asignarle los metadatos que componen la información estructurada de la anotación.

Suponemos que se está trabajando con un documento HTML y se realiza una selección de texto como la que aparece a continuación con fondo más oscuro. El objetivo de la selección de texto es delimitar el fragmento de texto sobre el que se realizará la anotación en un futuro.

```
1 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2
3   * Duis orci tellus, dignissim ac laoreet sit amet, porttitor et
      purus
4
5 Mauris congue ultrices sodales. Vivamus dignissim tristique leo, sit
    amet posuere ipsum hendrerit id.
```

Ejemplo 5.6: Selección de texto simple.

El código fuente HTML del documento en cuestión es el que aparece a continuación.

```
1 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
2 <br>
3 <ul>
4   <li>Duis orci tellus, dignissim ac laoreet sit amet, porttitor
      et purus. </li>
5 </ul>
6 <br>
7 <p>Mauris congue ultrices sodales. Vivamus dignissim tristique leo,
    sit amet posuere ipsum hendrerit id.
8 </p>
```

Ejemplo 5.7: Código HTML de la selección de texto.

En el ejemplo 5.7 anterior se muestra una de las selecciones de texto más simples que se pueden realizar. Esto es debido a que el fragmento de texto seleccionado está dentro de una misma etiqueta HTML (<li>) como se puede apreciar en su código fuente. En esta situación, el proceso de asignar un identificador único al fragmento de texto seleccionado es fácil: Se consigue incluyéndolo dentro de un elemento HTML tipo SPAN al que se le asignará un identificador y los atributos (RDFa) que se estimen oportunos. El resultado

se puede observar a continuación en el ejemplo 5.8.

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
<br>
<ul>
   <li>Duis orci tellus, <span id="654" class="annot">
        dignissim ac laoreet sit amet</span> ,porttitor et purus.
   </li>
</ul>
<br>
<p>Mauris congue ultrices sodales. Vivamus dignissim tristique leo,
    sit amet posuere ipsum hendrerit id.
</p>
```

Ejemplo 5.8: Delimitación e identificación de la zona de selección de texto.

Se puede observar en el ejemplo 5.8 cómo el fragmento de texto, con fondo más oscuro, ha sido rodeado por una etiqueta SPAN y que, además, a éste se le ha asignado un identificador único numérico y una clase CSS llamada `annot` para conseguir una visualización diferente en el navegador.

```
<div xmlns:r="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:a="http://www.w3.org/2000/10/annotation-ns#"
        xmlns:t="http://www.w3.org/2000/10/annotationType#"
        xmlns:dc="http://purl.org/dc/elements/1.1/">
   <span typeof="a:Annotation" about="http://ex.org/p.htm#654"/>
   <span resource="http://ex.org/p.htm" rel="a:Annotates"
            about="http://ex.org/p.htm#654"/>
   <span property="dc:creator"
            about="http://ex.org/p.htm#654">admin</span>
   <span property="a:Created"
            about="http://ex.org/p.htm#654">20/7/2009</span>
</div>
```

Ejemplo 5.9: Anotación Semántica en RDFa.

Una vez realizada la identificación del fragmento de texto que se desea anotar tiene lugar el proceso de anotación en lenguaje RDFa. En el listado 5.9 se puede observar un ejemplo de anotación semántica sobre el identificador del ejemplo 5.8 anterior, cuyo código se incrustaría en el documento original.

5.7.3. Delimitación e Identificación Multietiqueta

En este apartado se estudian los problemas que aparecen con una selección de texto multi-etiqueta como la que aparece a continuación en el ejemplo 5.10.

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    * Duis orci tellus, dignissim ac laoreet sit amet,
porttitor et purus Mauris congue ultrices sodales. Vivamus
    dignissim tristique leo, sit amet posuere ipsum hendrerit id.
```

Ejemplo 5.10: Selección de texto multietiqueta.

El código fuente HTML del documento en cuestión para esta nueva selección es el que aparece a continuación en el ejemplo 5.11.

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
<br>
<ul>
   <li>Duis orci tellus, dignissim ac laoreet sit amet,
   porttitor et purus. </li>
</ul>
<br>
<p>Mauris congue ultrices sodales. Vivamus dignissim tristique leo,
    sit amet posuere ipsum hendrerit id.
</p>
```

Ejemplo 5.11: Código HTML de la selección de texto multietiqueta.

Se puede observar en el código fuente HTML cómo la selección de texto delimitada en sombreado atraviesa los límites de varias etiquetas HTML. Este hecho ocasiona problemas a la hora de intentar delimitarlo y asignarle un identificador único, siguiendo las misma técnica del apartado 5.7.2.

```
<p>Lorem ipsum dolor sit amet,
    <span id="654">consectetur adipiscing elit.</p>
<br>
<ul>
   <li>Duis orci tellus, dignissim ac laoreet sit
   amet, porttitor et purus.</li>
</ul>
<br>
<p>Mauris congue ultrices sodales.</span> Vivamus dignissim
   tristique leo, sit amet posuere ipsum hendrerit id.
</p>
```

Ejemplo 5.12: Delimitación de la selección multietiqueta no válida.

En el ejemplo 5.12 se puede apreciar que, al tratarse de un fragmento de texto que recorre varias etiquetas HTML y en algunas falta la marca de inicio de

etiqueta o la de fin, la delimitación e identificación de éste mediante el uso de una única etiqueta SPAN produce código HTML no válido. Se observa, en principio, que se necesita usar una etiqueta SPAN en cada cambio de etiqueta HTML por la que discurre el fragmento de texto.

```
<p>Lorem ipsum dolor sit amet,
    <span id="654">consectetur adipiscing elit.</span></p>
<br>
<ul>
    <li><span id="654">Duis orci tellus, dignissim ac laoreet sit
    amet, porttitor et purus.</span></li>
</ul>
<br>
<p><span id="654">Mauris congue ultrices sodales.</span> Vivamus
    dignissim tristique leo, sit amet posuere ipsum hendrerit id.
</p>
```

Ejemplo 5.13: Delimitación de la selección multietiqueta no válida.

Si se realiza un proceso descrito en el párrafo anterior, se obtiene el resultado que aparece en el ejemplo 5.13. Se aprecia que la solución es válida en cuanto a la delimitación del fragmento de texto y a la validez del código HTML resultante, pero aparece el problema de que los identificadores deben de ser únicos y en el ejemplo aparecen duplicados.

La solución adoptada, a la hora de establecer la técnica de marcado de rangos flexibles de texto, pasa por la asignación de un identificador global que hará referencia a toda la selección de texto y el uso de identificadores locales para los fragmentos que forman los distintos elementos HTML que aparecen en la selección de texto. Se trata de una definición de un todo y de sus partes.

```
<p>Lorem ipsum dolor sit amet,
    <span id="654-1">consectetur adipiscing elit.</span></p>
<br>
<ul>
    <li><span id="654-2">Duis orci tellus, dignissim ac laoreet sit
        amet, porttitor et purus.</span></li>
</ul>
<br>
<p><span id="654-3">Mauris congue ultrices sodales.</span> Vivamus
    dignissim tristique leo, sit amet posuere ipsum hendrerit id.
</p>
```

Ejemplo 5.14: Delimitación de la selección multietiqueta no válida.

En el ejemplo del listado 5.14 anterior se puede observar cómo sería la identificación única de los distintos fragmentos que forman parte de la selección de texto de usuario. En este caso sí se trata de una identificación unívoca válida. Obsérvese que no aparece un identificador global (654) de la anotación que se está realizando; esta tarea se realiza a continuación en RDFa.

```
<span id="654" about="http://w3.ex.org/p.htm#654" rel="rdf:Seq">
      <span rel="rdf:li" resource="http://w3.ex.org/p.htm#654-1"/>
      <span rel="rdf:li" resource="http://w3.ex.org/p.htm#654-2"/>
      <span rel="rdf:li" resource="http://w3.ex.org/p.htm#654-3/>
</span>
```

Ejemplo 5.15: Definición del fragmento en RDFa.

En el ejemplo del listado 5.15 se ilustra cómo se realiza la definición del fragmento de texto en lenguaje RDFa mediante el uso del elemento contenedor *"Seq"* que proporciona el lenguaje RDF, el cual sirve para definir una lista ordenada de valores. En este caso en particular, se usa para describir los fragmentos que componen la anotación semántica.

```
<p>Lorem ipsum dolor sit amet, <span id="654-1">consectetur
    adipiscing elit. </span></p>
<ul><li><span id="654-2">Duis orci tellus, dignissim ac laoreet sit
    amet, porttitor et  purus.</span></li></ul>
<p><span id="654-3">Mauris congue ultrices sodales.</span> Vivamus
    dignissim tristique leo, sit amet posuere ipsum hendrerit id.
</p>
<div xmlns:r="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:a="http://www.w3.org/2000/10/annotation-ns#"
        xmlns:t="http://www.w3.org/2000/10/annotationType#"
        xmlns:dc="http://purl.org/dc/elements/1.1/">
   <span typeof="a:Annotation"
         about="http://ex.org/p.htm#654"></span>
   <span resource="http://ex.org/p.htm" rel="a:Annotates"
         about="http://ex.org/p.htm#654"></span>
   <span content="admin" property="dc:creator"
         about="http://ex.org/p.htm#654"></span>
   <span content="20/7/2009" property="a:Created"
         about="http://ex.org/p.htm#654"></span>
   <span id="654" about="http://w3.ex.org/p.htm#654" rel="rdf:Seq">
      <span rel="rdf:li" resource="http://w3.ex.org/p.htm#654-1"/>
      <span rel="rdf:li" resource="http://w3.ex.org/p.htm#654-2"/>
      <span rel="rdf:li" resource="http://w3.ex.org/p.htm#654-3"/>
   </span>
</div>
```

Ejemplo 5.16: Documento HTML con anotaciones incrustadas en RDFa.

Para concluir la descripción de esta técnica, se muestra en el ejemplo 5.16 cómo quedaría el documento en su conjunto, con la anotación semántica incrustada en el mismo. En la sección 5.4 se comentó que el sistema FLERSA utiliza un componente que realiza las funciones de repositorio de contenidos, encargado de almacenar tanto el contenido web como la información semántica referida a éste (anotaciones semánticas). Es posible que la herramienta específica que se use para implementar el repositorio de contenidos disponga de motor de inferencia y de capacidades de procesamiento de consultas SparQL. Se plantea entonces la cuestión de que se necesita que las anotaciones semánticas estén expresadas en lenguaje RDF, en lugar de usar la sintaxis RDFa, para obtener así los beneficios que este tipo de herramientas ofrecen. La solución pasa por utilizar procesadores de RDFa y traducir las anotaciones semánticas a lenguaje RDF.

```
<rdf:RDF>
  <a:Annotation rdf:about="http://w3.ex.org/p.htm#654">
    <rdf:Seq>
      <rdf:Description>
        <rdf:li>
          <a:Annotation rdf:about="http://w3.ex.org/p.htm#654-1">
            <a:Body> consectetur adipiscing elit. </a:Body>
          </a:Annotation>
        </rdf:li>
        <rdf:li>
          <a:Annotation rdf:about="http://w3.ex.org/p.htm#654-2">
            <a:Body>Duis orci tellus, dignissim ac laoreet sit amet,
                porttitor et purus.</a:Body>
          </a:Annotation>
        </rdf:li>
        <rdf:li>
          <a:Annotation rdf:about="http://w3.ex.org/p.htm#654-3">
            <a:Body> Mauris congue ultric sodales.</a:Body>
          </a:Annotation>
        </rdf:li>
      </rdf:Description>
    </rdf:Seq>
    <a:Annotates rdf:resource="http://w3.ex.org/p.htm"/>
      <dc:creator>admin</dc:creator>
      <a:Created>20/7/2009</a:Created>
  </a:Annotation>
  <rdf:Description rdf:about="http://w3.ex.org/p.htm">
    <dc:date.modified>2009-07-24</dc:date.modified>
    <dc:date>2009-07-24 19:44:45</dc:date>
    <dc:title> Lorem Ipsum</dc:title>
  </rdf:Description>
</rdf:RDF>
```

Ejemplo 5.17: Traducción a RDF del ejemplo del listado 5.16.

Por ejemplo, se podría traducir a RDF el ejemplo 5.16 expresado en lenguaje RDFa mediante un procesador como por ejemplo RDFa Distiller[2]. La semántica expresada en los ejemplos 5.16 y 5.17, es idéntica.

5.7.4. Proceso de Marcado de Rangos Flexibles

A continuación, se va a realizar una descripción del proceso de marcado según la técnica de definición de rangos flexibles de texto, para su anotación, explicada en las secciones 5.7.2 y 5.7.3 anterior. El objetivo del proceso es delimitar e identificar la selección de texto del documento web que el usuario desea anotar. Se compone de las siguientes etapas:

- El proceso parte de la selección por parte del usuario de un fragmento de texto en un documento web.
- Se define un identificador global de marcado y, a continuación, se distinguen dos casuísticas:
  - En el caso en que la selección de texto afecte a varias etiquetas HTML, es necesario realizar una identificación unívoca de las distintas etiquetas que, a modo de fragmentos, componen el texto seleccionado para la anotación. Para ello, se realiza una delimitación de cada uno de los elementos HTML mediante el uso del elemento SPAN de HTML. Una vez que los elementos HTML han sido delimitados con etiquetas SPAN, se consideran *fragmentos* y es necesario identificarlos. El criterio de identificación es usar el identificador de marca global seguido de la numeración del fragmento. Para terminar, se establecerá la relación de pertenencia de los fragmentos al identificador global de marcado haciendo uso de un contenedor *"RDF:Seq"*.
  - En el caso de tratarse de una selección de texto simple, se delimitará el texto mediante etiquetas SPAN de HTML y se asignará un identificador de marcado global directamente a dicha etiqueta.

  Llegados a este punto, queda resuelto el proceso de identificación unívoca del texto seleccionado por el usuario de alguna de las dos formas anteriores y se está en disposición de asociar al marcado, mediante lenguaje RDFa, los metadatos que componen el contenido de la anotación semántica relacionándolos con el identificador global de la marca.

En la figura 5.9 podemos observar un diagrama de flujo del proceso de aplicación de la técnica descrita en este apartado.

[2] `http://www.w3.org/2007/08/pyRdfa`

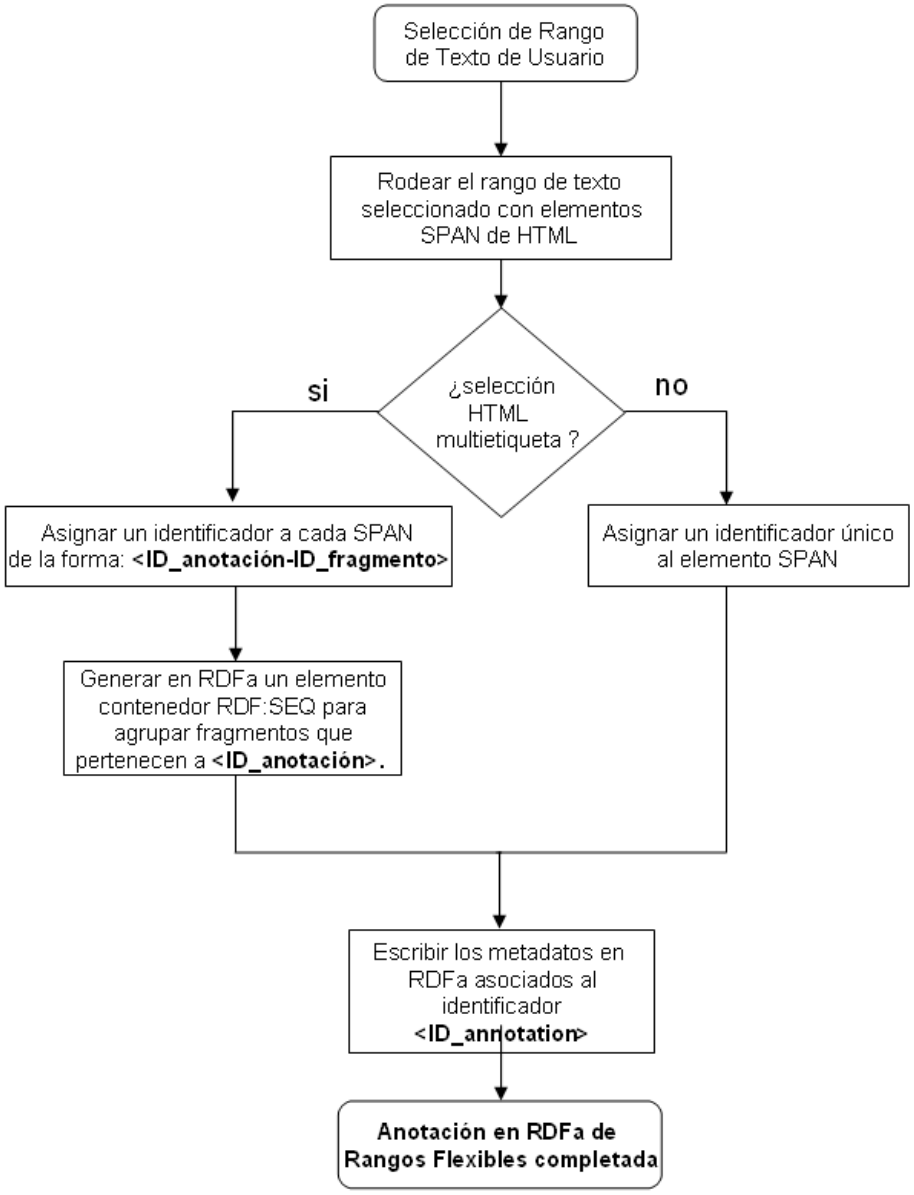


Figura 5.9: Diagrama de flujo del proceso de marcado rangos flexibles.

5.8. Proceso Manual de Anotación Semántica

El proceso manual de anotación semántica fue presentado en el artículo de Navarro-Galindo y Samos (2010b). En esta sección se revisa el proceso presentado, incluyendo ejemplos y describiendo cómo enriquecer las anota-

ciones semánticas que siguen al proceso de marcado.

El proceso se basa en el uso de la técnica de definición de rangos flexibles explicada en la sección 5.7. Centrándonos en la funcionalidad que ofrece el sistema FLERSA para la anotación manual de contenido web, se presenta cómo tiene lugar el proceso de anotación. Los pasos son los siguientes:

1. Selección del fragmento de texto sobre el cual desea realizar la anotación, evitando que le afecten las etiquetas HTML asociadas.

```
<p>Lorem ipsum dolor sit amet,
    consectetur adipiscing elit.</p>
<br>
<ul>
   <li>Duis orci tellus, dignissim ac laoreet sit amet,
       porttitor et purus. </li>
</ul>
<br>
<p>Mauris congue ultrices sodales. Vivamus dignissim tristique
    leo, sit amet posuere ipsum hendrerit id.
</p>
```

Ejemplo 5.18: Código HTML correspondiente a una selección.

En el ejemplo 5.18 se muestra el código fuente HTML correspondiente a un hipotético fragmento de texto que se pretende anotar. Se puede observar en él cómo la selección de texto, delimitada con fondo gris, atraviesa los límites de varias etiquetas HTML. Este hecho nos ocasiona un problema a la hora de intentar delimitarlo con un identificador único.

2. Marcado del fragmento de texto seleccionado en lenguaje HTML. Se genera un identificador global que se usa para el marcado del fragmento de texto seleccionado y la posterior definición de la anotación semántica. También es necesario seguir una estrategia que permita la identificación local de los elementos HTML que pertenecen al fragmento de texto seleccionado y la definición de la relación de pertenencia. El identificador global se usará posteriormente, como punto de referencia (anchor-anclaje) en el resto de labores de anotación semántica. Los identificadores globales serán usados para asociarles metadatos. Los identificadores locales también posibilitarán la delimitación visual del fragmento de texto seleccionado.

```
<p>Lorem ipsum dolor sit amet,
    <span id="654-1">consectetur adipiscing elit.</span></p>
<br>
<ul>
   <li><span id="654-2">Duis orci tellus, dignissim ac laoreet sit amet,
        porttitor et purus.</span></li>
</ul>
<br>
<p><span id="654-3">Mauris congue ultrices sodales.</span>
   Vivamus dignissim tristique leo, sit amet posuere ipsum
   hendrerit id.
</p>
```

Ejemplo 5.19: Marcado de un fragmento de texto.

En el ejemplo 5.19 se ha realizado un marcado de un fragmento de texto. Se puede observar que el identificador de fragmento global es 654, y que a su vez se compone de tres subfragmentos de texto con identificadores locales 654-1, 654-2 y 654-3. Para una descripción más detallada véase la sección 5.7.

3. Inclusión de las sentencias que componen la anotación semántica base. Se generan las sentencias necesarias para definir una nueva instancia o individuo del concepto anotación definido en la ontología FLERSA-Ontology. Cada una de las sentencias describe uno de los atributos estudiados en la sección 5.6. Las sentencias pueden expresarse bien en lenguaje RDF y almacenarse en la Base de Conocimiento del Sistema de Anotación, o expresarse en RDFa y almacenarse de forma incrustada en el mismo documento que se anota.

```
<div xmlns:r="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:a="http://www.w3.org/2000/10/annotation-ns#"
    xmlns:t="http://www.w3.org/2000/10/annotationType#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">

    <span typeof="a:Annotation"
          about="http://ex.org/p.htm#654"></span>
    <span resource="http://ex.org/p.htm" rel="a:Annotates"
          about="http://ex.org/p.htm#654"></span>
    <span content="José Luis Navarro" property="dc:creator"
          about="http://ex.org/p.htm#654"></span>
    <span content="20/7/2009" property="a:Created"
          about="http://ex.org/p.htm#654"></span>
</div>
```

Ejemplo 5.20: Anotación semántica incrustada en HTML.

Se puede observar en el ejemplo 5.20 cómo almacenan de forma incrustada las sentencias que componen una anotación semántica. Su significado es el siguiente: En las líneas 6 y 7 se define una instancia de la clase "*Annotation*" con identificador "654"; en las líneas 8 y 9 se establece el valor la propiedad "*Annotates*", indicando que la anotación con identificador "654" está situada dentro de la página web "http://ex.org/p.htm"; en las líneas 10 y 11 se usa la propiedad "*creator*", perteneciente al vocabulario Dublín Core, para establecer el creador de la anotación que en este caso es "José Luis Navarro"; finalmente, en las líneas 12 y 13 se establece el valor de la propiedad "*Created*" para indicar la fecha de creación de la anotación.

Hasta aquí se describe el proceso que permite definir anotaciones semánticas básicas. En principio, las anotaciones no proporcionan ninguna ventaja semántica por sí mismas, son anotaciones simples en las que sólo se definen propiedades básicas. Es necesario un proceso extra de edición y asociación a ontologías que permita enriquecerlas y dotarlas de funcionalidad semántica. Las posibilidades que ofrece FLERSA-ontology son:

- Añadir sentencias que describan de qué temática se habla en el fragmento de texto asociado. Para realizar esta tarea se utiliza la propiedad *"Related"* y se usan como valores los conceptos que proporcionan las taxonomías de la Base de Conocimiento. Esta característica nos va a permitir en el futuro realizar búsquedas inteligentes aprovechando las relaciones de especialización/generalización existente en la organización jerárquica de los conceptos descritos en las taxonomías y la capacidad de inferencia de las ontologías.

```
1  <rdf:RDF xmlns:a="http://www.scms.es/Annotation#"
2           xmlns:auto="http://www.scms.es/automobile#"
3           xmlns:dc="http://purl.org/dc/elements/1.1/">
4   <a:Annotation rdf:about="http://site/Webpage.htm#1001054">
5     <a:Annotates rdf:resource="http://site/Webpage.htm"/>
6     <a:Context rdf:resource="http://site/Webpage.htm#1001054"/>
7     <a:Body> This produced economical engines with earlier four-
          cylinder designs rated at 40 horsepower, compared with
          the large volume V-8 American engines</a:Body>
8     <a:Author>José Luis Navarro</a:Author>
9     <a:Granularity rdf:resource="a:Paragraph"/>
10    <a:Related rdf:resource="auto:engine"/>
11    <a:Created >2009-10-01 19:44:45</a:Created>
12  </a:Annotation>
13 </rdf:RDF>
```

Ejemplo 5.21: Anotación RDF según la estructura Annotea.

En el ejemplo 5.21 se puede observar la definición de una anotación semántica con identificador número "1001054" (línea 4) definido en la página web "http://site/Webpage.htm" (línea 5, propiedad *"annotates"*); el contexto de creación de la anotación (línea 6, propiedad *"context"*) es "http://site/Webpage.htm#1001054"; el texto al que la información semántica se refiere es un párrafo (línea 9, propiedad *"granularity"*) cuyo contenido trata sobre el concepto "engine" (línea 10, propiedad *"related"*); también aparece otra información menos relevante como: El autor y la fecha de creación (líneas 8 y 11).

- Definición de individuos pertenecientes a distintos conceptos en base a las anotaciones semánticas realizadas en un documento. Inicialmente es necesario realizar anotaciones básicas dentro de un documento que actuarán como puntos de referencia. A continuación se define el tipo del concepto del que habla el documento haciendo uso de la propiedad *"type"* de RDF-Schema. Por último, se utiliza el vocabulario oportuno dentro del dominio del concepto para describir al individuo.

```
1  <rdf:RDF xmlns:a="http://www.scms.es/Annotation#"
2           xmlns:auto="http://www.scms.es/automobile#"
3           xmlns:dc="http://purl.org/dc/elements/1.1/">
4     <a:Annotation rdf:about="http://w.ex.com/pag.htm#1001120">
5        <a:Granularity rdf:resource="a:Pharagraph"/>
6        <a:Created >19/9/2009</a:Created>
7        <a:Annotates rdf:resource="http://w.ex.com/pag.htm"/>
8        <a:Author>admin</a:Author>
9        <a:Body>José Luis Navarro</a:Body>
10       <a:Section rdf:resource="a:Text"/>
11       <a:Context rdf:resource="http://w.ex.com/pag.htm#1001120"
            />
12    </a:Annotation >
13
14    <foaf:Person rdf:about="http://w.ex.com/pag.htm ">
15       <dc:title> prueba</dc:title>
16       <foaf:Name rdf:resource="http://w.ex.com/pag.htm
             #1001120"/ >
17       <dc:date>2009-10-19 19:30:57</dc:date>
18       <dc:creator>Administrator</dc:creator>
19    </foaf:Person>
20 </rdf:RDF>
```

Ejemplo 5.22: Definición de individuos en base a anotaciones semánticas.

Se puede observar en el ejemplo 5.22 cómo, la anotación semántica número "1001120", sirve para establecer una referencia hacia el fragmento de texto que delimita el nombre de una persona (línea 7). A continuación, en la parte inferior del ejemplo, se puede apreciar cómo se ha definido un individuo

del concepto *"foaf:Person"* que usa en su propiedad *"foaf:Name"* la referencia creada en la anotación anterior. El estudio del resto de propiedades carece de interés en este ejemplo ya que son muy similares a las estudiadas anteriormente en otros ejemplos.

5.9. Proceso Automático de Anotación Semántica

El proceso automático de anotación semántica también fue presentado en el artículo de Navarro-Galindo y Samos (2010b). En esta sección se revisa el proceso presentado, describiéndose por medio de diagramas cómo tiene lugar dicho proceso.

El objetivo que se marca en este proceso es usar la infraestructura de marcado y anotación semántica de contenidos web, combinada con técnicas de aprendizaje (Machine Learning), para facilitar el proceso de anotación, consiguiendo su automatización.

La creación de anotaciones semánticas automática es una característica clave en el sistema FLERSA. Gracias a esta característica, los ingenieros del conocimiento son capaces de entrenar al sistema para que, automáticamente, se establezcan relaciones entre las anotaciones semánticas de un documento y los conceptos que proporcionan distintas taxonomías de la Base de Conocimiento del sistema.

En el sistema FLERSA, se ha desarrollado un enfoque híbrido que combina dos técnicas de aprendizaje automático bien conocidas: El modelo de espacio vectorial y los n-gramas.

5.9.1. Principios Teóricos

En el **Modelo de Espacio Vectorial** se considera que cada documento, perteneciente a una colección, es un vector de pesos en un espacio vectorial de T dimensiones, donde T es el número de términos diferentes que aparecen en la colección.

$$D_i = (d_{i1}, d_{i2}, d_{i3}, ..., d_{in}). \tag{5.1}$$

donde d_{ij} es el peso del término j-ésimo para el documento D_i

En el modelo de espacio vectorial, cuando se calcula el peso de un término en un documento, se tienen en cuenta los siguientes aspectos:

- La frecuencia de ocurrencia de un término en un documento, *tf* (Luhn, 1958). Los términos más repetidos dentro de un documento son, en principio, más relevantes que los menos usados.

- El número de documentos en la colección en los cuales el término aparece, *df* (Sparck Jones, 1988).
- Los términos comunes en varios documentos serán menos relevantes que los poco comunes.
- La longitud del documento, para garantizar que todos los documentos se comportan de manera similar, independientemente de su longitud. En otras palabras, no existe una relación entre la relevancia de un documento y su longitud.

El modelo de espacio vectorial se define como sigue a continuación:

- Se genera un vector de pesos para cada documento, de manera que el valor del término j-ésimo que aparece en el documento i-ésimo de una colección se calcula usando la ecuación 5.2.

$$w_{i,j} = tf_{i,j} * log\left(\frac{D}{df_j}\right). \tag{5.2}$$

  donde:

  - $tf_{i,j}$ es el número de veces que el término j-ésimo aparece en el documento i-ésimo.
  - df_j es el número de documentos que contienen el término j-ésimo.
  - D es el número total de documentos considerados.

- Dada una consulta, es posible medir la similitud entre el vector de consulta y los vectores pertenecientes a la colección usando la ecuación coseno 5.3.

$$Sim(Q, D_i) = \frac{\sum_i w_{Q,j} w_{i,j}}{\sqrt{\sum_j w_{Q,j}^2}\sqrt{\sum_i w_{i,j}^2}}. \tag{5.3}$$

Como se puede observar, la definición del modelo de espacio vectorial no requiere de cálculos excesivamente complejos. Es por esta razón por lo que se ha pensado en este modelo, porque su implementación en un sistema no resultaría, en principio, muy costosa. Además, se sabe que el modelo se usa normalmente para la categorización de texto, dado que los sistemas basados en él pueden entrenarse fácilmente. Sin embargo, tiene varias limitaciones:

1. Costoso en términos de cálculo intenso.
2. Lento desde el punto de vista computacional, ya que requiere mucho tiempo de procesamiento.

3. Cada vez que se añade un nuevo término al espacio de términos, se necesita recalcular todos los vectores de pesos.

4. Presenta coincidencias falso negativo y/o falso positivo cuando se trabaja con documentos de contenido similar o que usan impropiamente sus palabras.

A pesar de las desventajas que presenta el modelo, se ajusta perfectamente al proceso automático de anotación semántica y, en particular, con la tarea de determinar el concepto del que trata un fragmento de texto que esté siendo marcado.

En un S-CMS existen conceptos en lugar de categorías y existen anotaciones (sobre fragmentos de texto) en lugar de documentos, pero el problema es en esencia el mismo. Por lo tanto, se ha adoptado el modelo de espacio vectorial en el sistema FLERSA, resolviendo las desventajas que presenta de la siguiente forma:

1. **Cálculo intensivo**. Debido a que el cálculo de la ecuación 5.3 es costoso en términos de tiempo de computación, en el sistema FLERSA se puede precomputar el elemento denominador que es independiente del vector de consulta, proporcionando así tiempos de respuesta adecuados para los usuarios que utilizan el sistema de anotación automatizado.

2. **Cambio en los términos**. La herramienta sigue un enfoque "paga según recibas" (Halevy et al., 2006). Los términos cambian sólo de vez en cuando. El ingeniero de conocimiento será el responsable de proporcionar un Corpus básico desde el cual comenzar a trabajar. Gradualmente, pueden incorporarse nuevo términos al sistema conforme el contenido web crece. El recálculo de todos los vectores sólo es necesario cuando el ingeniero de conocimiento valide las nuevas anotaciones susceptibles de ser incorporadas al Corpus. El proceso de recálculo será ejecutado explícitamente por el ingeniero del conocimiento que administra el sistema desde la aplicación de administración del mismo. Cuanto más tiempo se emplee en ajustar el sistema (tuning), mejores resultados se obtendrán.

3. **Coincidencias falsas**. El modelo de espacio vectorial no especifica qué elementos de un documento deben ser usados como términos. Esto es, el uso de palabras no es obligatorio y se admiten otras posibilidades tales como el uso de n-gramas. El sistema FLERSA usa n-gramas para asegurar una tasa baja de errores a la hora de determinar el concepto del que se trata en un fragmento de texto durante el proceso de anotación semántica.

En el contexto de computación lingüística, un **n-grama** es una subsecuencia de n-elementos pertenecientes a una secuencia dada. Pueden referirse a fonemas, sílabas, letras, palabras o pares dependiendo de la aplicación.

En esta tesis, el concepto n-grama se refiere a una secuencia continua de caracteres alfabéticos separada de otras secuencias mediante espacios o signos de puntuación. Un n-grama, entonces, coincide con lo que comúnmente llamamos "palabra" en su dimensión ortográfica.

La letra *"n"* en el término *"n-grama"* es un símbolo matemático que se usa para representar la serie completa de los números naturales (1,2,3 ...). Los tres tipos de n-gramas que se consideran en esta tesis son: "Monogramas", también llamados "unigramas" o "1-gramas" (por ejemplo "white"), y se refieren a palabras individuales; los "bigramas" o "2-gramas" son cadenas de dos palabras (por ejemplo "white blood"); por último, los "trigramas" o "3-gramas" son formaciones de tres palabras (por ejemplo "white blood cells").

5.9.2. Enfoque del Modelo Híbrido

La mayoría de las técnicas de aprendizaje, incluido el Modelo de Espacio Vectorial, se basan en cálculos estadísticos de las distintas frecuencias del léxico (como por ejemplo el número de ocurrencias de una palabra) para estimar el concepto del que trata un documento, bajo la suposición de que estas estadísticas del léxico son lo suficientemente representativas del contenido informativo de un documento. Este tipo de estimaciones asumen que las palabras aparecen independientemente las unas de las otras, ignorando la semántica composicional del lenguaje y causando diversos problemas, tales como la ambigüedad en la comprensión de la información textual, mala interpretación de la intención informativa original y limitación del ámbito semántico del texto.

Nuestro enfoque híbrido propone el uso de estadísticas léxicas, tales como el Modelo de Espacio Vectorial, combinado y enriquecido con n-gramas, con el objetivo de mejorar la estimación del concepto del que trata un documento, evitando los problemas heredados del Modelo de Espacio Vectorial, explicados anteriormente, los cuales causan resultados erróneos tales como falsos positivos y falsos negativos.

La propuesta de utilizar un modelo híbrido **Modelo Espacio Vectorial + n-gramas** queda justificada por las siguientes razones:

- Está demostrada la efectividad de aplicar el modelo de espacio vectorial para resolver problemas de categorización de texto. Dado que los sistemas de anotación semántica trabajan con fragmentos de texto sobre

los que se describe información semántica, pensamos que este modelo se ajusta a la perfección para ayudar en las tareas de clasificación de los fragmentos de texto dentro de taxonomías de conceptos y, posteriormente, incorporar esta información en sus anotaciones semánticas.

- Vistos los principios teóricos del modelo, se valora positivamente la facilidad con que éstos pueden ser llevados a la práctica mediante la implementación del modelo en un S-CMS.
- El uso de n-gramas de distintos niveles (monogramas, bigramas, trigramas) a modo de términos, proporciona al modelo mayor efectividad a la hora de realizar las categorizaciones de texto.

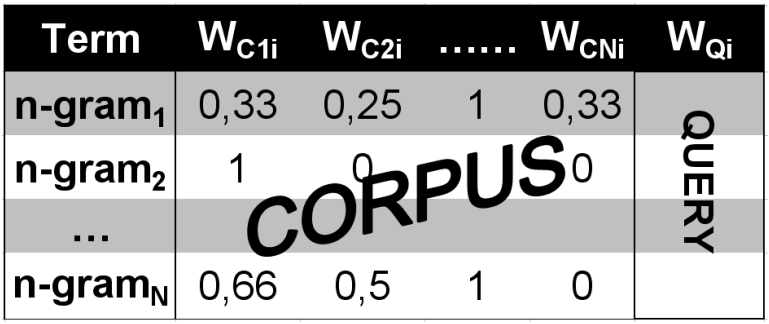


Figura 5.10: Esquema de representación del modelo de espacio vectorial.

El esquema computacional desarrollado en FLERSA para la implementación de nuestro modelo híbrido es el que se muestra en la figura 5.10, sobre el que hacemos las consideraciones siguientes:

- Los pesos se calculan siguiendo la ecuación 5.2.
- La columna W_{Ci} muestra el peso del concepto "C" para el término i-ésimo.
- La columna W_{Qi} muestra el peso de la consulta "Q" para el término i-ésimo.

Cuando se construye un modelo híbrido aplicado a un área específica, se necesita un Corpus de textos para cada uno de los conceptos que se pretendan modelar. Entonces, se extraen los n-gramas (monogramas, bigramas y trigramas) a partir de los textos del Corpus y se calculan los pesos de acuerdo con la ecuación 5.2 anterior. Finalmente, se calcula la ecuación de similitud 5.3 para cada par concepto-consulta. Se realiza un análisis de similitud separado para monogramas, bigramas y trigramas; sus valores se ponderan de acuerdo con la ecuación 5.4.

$$\begin{aligned} Sim_{Global}(Q, C_i) =& \alpha \cdot Sim_{Tri}(Q, C_i) + \beta \cdot Sim_{Bi}(Q, C_i) \\ &+ \gamma \cdot Sim_{Mono}(Q, C_i). \\ & siendo\, \alpha + \beta + \gamma = 1. \end{aligned} \tag{5.4}$$

El estudio de los valores ideales de los parámetros que α, β y γ queda fuera del ámbito de la presente tesis. Actualmente se están usando los valores de referencia 0.7, 0.2 y 0.1 respectivamente, obteniéndose buenos resultados. Parece lógico que se le asigne más peso a la similitud de los trigramas dado que es mayor la dificultad de que un mismo trigrama aparezca en textos que pertenezcan a categorías distintas. Esto mimo ocurre, en menor medida, con los bigramas y por eso el peso que se le ha asignado a los mismos es el doble que el peso de los monogramas.

5.9.3. Caso Práctico

Este apartado ilustra el funcionamiento del proceso de categorización de textos siguiendo el modelo híbrido presentado en ésta sección, mediante el desarrollo de un caso práctico donde se detallan los cálculos que se realizan a la hora de determinar la categoría de un fragmento de texto.

Suponiendo que se dispone de tres textos pertenecientes a categorías distintas, identificadas como C1, C2 y C3, y que también se dispone de un texto consulta, identificado como Q, del cual se va a estudiar la similaridad del mismo con respecto a las categorías. Para simplificar el proceso se han eliminado los signos de puntuación y nexos de los mismos, ya que se tratan de palabras y símbolos que no aportan significado. Los textos son los que aparecen a continuación:

- **Blood (C1)**: It is circulating tissue composed fluid plasma cells red blood cells white blood cells platelets.
- **Digestive system (C2)**: It is responsible breakdown absorption various foods liquids needed sustain life.
- **Muscular system (C3)**: It is biological system humans produces movement.
- **Consulta (Q)**: White blood cells are part immune system are responsible fighting off desease.

En la tabla que aparece en la figura 5.11 puede observarse que aparecen diversas columnas, las cuales se han añadido para ilustrar todos los cálculos necesarios para conseguir el peso de los términos (monogramas). Analicemos los datos columna por columna:

	$tf_{i,j}$							$w_{i,j}=tf_{i,j}*log(D/df_j)$			
Término (j)	$tf_{C1,j}$	$tf_{C2,j}$	$tf_{C3,j}$	$tf_{Q,j}$	df_j	D/df_j	$log(D/df_j)$	$W_{C1,j}$	$W_{C2,j}$	$W_{C3,j}$	$W_{Q,j}$
absorption		1			1	3	0,48		0,48		
biological			1		1	3	0,48			0,48	
blood	2			1	1	3	0,48	0,95			0,48
breakdown		1			1	3	0,48		0,48		
cells	3			1	1	3	0,48	1,43			0,48
circulating	1				1	3	0,48	0,48			
composed	1				1	3	0,48	0,48			
fluid	1				1	3	0,48	0,48			
foods		1			1	3	0,48		0,48		
humans			1		1	3	0,48			0,48	
is	1	1	1		3	1	0,00				
it	1	1	1		3	1	0,00				
life		1			1	3	0,48		0,48		
liquids		1			1	3	0,48		0,48		
movement			1		1	3	0,48			0,48	
needed		1			1	3	0,48		0,48		
plasma	1				1	3	0,48	0,48			
platelets	1				1	3	0,48	0,48			
produces			1		1	3	0,48			0,48	
red	1				1	3	0,48	0,48			
responsible		1		1	1	3	0,48		0,48		0,48
sustain		1			1	3	0,48		0,48		
system			1	1	1	3	0,48			0,48	0,48
tissue	1				1	3	0,48	0,48			
various		1			1	3	0,48		0,48		
white	1			1	1	3	0,48	0,48			0,48

Figura 5.11: Representación de los monogramas de los textos según el modelo de espacio vectorial.

- Columnas 1 - 5: Se construye un índice de los términos que aparecen en los textos de las categorías C1, C2 y C3. Se determina la frecuencia de los términos (tf) para cada categoría y para la consulta Q.

- Columnas 6 - 8: Se calcula la frecuencia de aparición de cada término en las categorías. Se calcula los componentes de la ecuación 5.2.

- Columnas 9 - 12: Se toman los componentes de la ecuación 5.2 calculados en las columnas anteriores para calcular los pesos de los términos. Obsérvese cómo los términos con mayor frecuencia de aparición en una única categoría, como por ejemplo los términos "*blood*" y "*cells*", obtienen mayor peso que los que aparecen simultáneamente en todas las categorías. Estas columnas pueden verse como una matriz dispersa en la que la mayoría de entradas son iguales a cero.

Puntualizar que en la figura 5.11 se han omitido los valores iguales a cero para dar mayor claridad a la tabla y proporcionar así una mejor comprensión del caso práctico.

En primer lugar, se realiza el análisis de similitud en monogramas, calculando el numerador (ecuación 5.5) y los dos componentes del denominador (ecuaciones 5.6 y 5.7) de la ecuación 5.3.

$$Q \cdot C_i = \sum_i w_{Q,j} * w_{Ci,j}. \tag{5.5}$$

$$Q_{Mono} \cdot C1_{Mono} = 0{,}48 \cdot 0{,}95 + 0{,}48 \cdot 1{,}43 + 0{,}48 \cdot 0{,}48 = 1{,}37$$

$$Q_{Mono} \cdot C2_{Mono} = 0{,}48 \cdot 0{,}48 = 0{,}23$$

$$Q_{Mono} \cdot C3_{Mono} = 0{,}48 \cdot 0{,}48 = 0{,}23$$

$$|C_i| = \sqrt{\sum_i w_{Ci,j}^2}. \tag{5.6}$$

$$|C1_{Mono}| = \sqrt{0{,}95^2 + 1{,}43^2 + (0{,}48^2)^8} = \sqrt{0{,}91 + 2{,}05 + 0{,}23^8} = 2{,}19$$

$$|C2_{Mono}| = \sqrt{(0{,}48^2)^9} = \sqrt{0{,}23^9} = 1{,}43$$

$$|C3_{Mono}| = \sqrt{(0{,}48^2)^5} = \sqrt{0{,}23^5} = 1{,}07$$

$$|Q| = \sqrt{\sum_j w_{Q,j}^2}. \tag{5.7}$$

$$|Q_{Mono}| = \sqrt{(0{,}48^2)^5} = \sqrt{0{,}23^5} = 1{,}07$$

Una vez calculados los componentes del numerador y denominador, se calcula el grado de similitud de la consulta Q con respecto a las categorías C1, C2 y C3 como sigue:

$$Sim_{Mono}(C1, Q) = \frac{Q_{Mono} \cdot C1_{Mono}}{|C1_{Mono}| \cdot |Q_{Mono}|} = \frac{1{,}37}{2{,}19 \cdot 1{,}07} = 0{,}58$$

$$Sim_{Mono}(C2, Q) = \frac{Q_{Mono} \cdot C2_{Mono}}{|C2_{Mono}| \cdot |Q_{Mono}|} = \frac{0{,}23}{1{,}43 \cdot 1{,}07} = 0{,}15$$

$$Sim_{Mono}(C3, Q) = \frac{Q_{Mono} \cdot C3_{Mono}}{|C3_{Mono}| \cdot |Q_{Mono}|} = \frac{0{,}23}{1{,}07 \cdot 1{,}07} = 0{,}2$$

Aquí concluye el cálculo de la similitud a nivel de monogramas. Obsérvense los valores de similitud de la consulta Q con respecto a las tres categorías. Se aprecia que entre la consulta Q y la categoría C1 es donde existe un mayor nivel de similitud; esto es debido a que en el texto de la categoría C1 y de la consulta Q se comparten los monogramas "white", "cells" y "blood", mientras que con las categorías C2 y C3 sólo se comparte un monograma con cada una: "Responsible" con C2 y "system" con C3.

En segundo lugar, se realiza el cálculo de la similitud a nivel de bigramas. Los bigramas se forman a partir de las palabras de los textos de las categorías, tomándolas de dos en dos. Por ejemplo, dado el texto siguiente: "*It is biological system humans produces movement*", se generarían los siguientes bigramas: "*It-is, is-biological, biological-system, system-humans, humans-produces, produces-movement*".

En la tabla que aparece en la figura 5.12 puede observarse los cálculos necesarios para conseguir el peso de los bigramas que aparecen en los textos de las categorías C1, C2 y C3.

Puntualizar que también en la figura 5.12 se han omitido los valores iguales a cero para dar mayor claridad a la tabla y proporcionar así una mejor comprensión del caso práctico.

El cálculo de la similitud para los bigramas se realiza de forma similar al cálculo de monogramas, esto es, calculando el numerador (ecuación 5.5) y los dos componentes del denominador (ecuaciones 5.6 y 5.7) de la ecuación 5.3 como sigue a continuación:

$$Q_{Bi} \cdot C1_{Bi} = 0{,}48 \cdot 0{,}95 + 0{,}48 * 0{,}48 = 0{,}68$$

$$Q_{Bi} \cdot C2_{Bi} = 0$$

$$Q_{Bi} \cdot C3_{Bi} = 0$$

$$|C1_{Bi}| = \sqrt{0{,}95^2 + (0{,}48^2)^1 1} = \sqrt{0{,}91 + 0{,}23^1 1} = 1{,}85$$

$$|C2_{Bi}| = \sqrt{(0{,}48^2)^9} = \sqrt{0{,}23^9} = 1{,}43$$

$$|C3_{Bi}| = \sqrt{(0{,}48^2)^5} = \sqrt{0{,}23^5} = 1{,}07$$

	$tf_{i,j}$							$w_{i,j}=tf_{i,j}*log(D/df_j)$			
Término (j)	$tf_{C1,i}$	$tf_{C2,i}$	$tf_{C3,i}$	$tf_{Q,i}$	df_i	D/df_i	$log(D/df_i)$	$W_{C1,i}$	$W_{C2,i}$	$W_{C3,i}$	$W_{Q,i}$
absorption-various		1			1	3	0,48		0,48		
biological-system			1		1	3	0,48			0,48	
blood-cells	2			1	1	3	0,48	0,95			0,48
breakdown-absorption		1			1	3	0,48		0,48		
cells-platelets	1				1	3	0,48	0,48			
cells-red	1				1	3	0,48	0,48			
cells-white	1				1	3	0,48	0,48			
circulating-tissue	1				1	3	0,48	0,48			
composed-fluid	1				1	3	0,48	0,48			
fluid-plasma	1				1	3	0,48	0,48			
foods-liquids		1			1	3	0,48		0,48		
humans-produces			1		1	3	0,48			0,48	
is-biological			1		1	3	0,48			0,48	
is-circulating	1				1	3	0,48	0,48			
is-responsible		1			1	3	0,48		0,48		
It-is	1	1	1		3	1	0,00				
liquids-needed		1			1	3	0,48		0,48		
needed-sustain		1			1	3	0,48		0,48		
plasma-cells	1				1	3	0,48	0,48			
produces-movement			1		1	3	0,48			0,48	
red-blood	1				1	3	0,48	0,48			
responsible-breakdown		1			1	3	0,48		0,48		
sustain-life		1			1	3	0,48		0,48		
system-humans			1		1	3	0,48			0,48	
tissue-composed	1				1	3	0,48	0,48			
various-foods		1			1	3	0,48		0,48		
white-blood	1			1	1	3	0,48	0,48			0,48

Figura 5.12: Representación de los bigramas de los textos según el modelo de espacio vectorial.

$$|Q_{Bi}| = \sqrt{(0{,}48^2)^2} = \sqrt{0{,}23^2} = 0{,}67$$

Una vez calculados los componentes del numerador y denominador, se calcula el grado de similitud de la consulta Q con respecto a las categorías C1, C2 y C3 como sigue:

$$Sim_{Bi}(C1, Q) = \frac{Q_{Bi} \cdot C1_{Bi}}{|C1_{Bi}| \cdot |Q_{Bi}|} = \frac{0{,}68}{1{,}85 \cdot 0{,}67} = 0{,}55$$

$$Sim_{Bi}(C2, Q) = \frac{Q_{Bi} \cdot C2_{Bi}}{|C2_{Bi}| \cdot |Q_{Bi}|} = \frac{0}{1{,}43 \cdot 0{,}67} = 0$$

$$Sim_{Bi}(C3, Q) = \frac{Q_{Bi} \cdot C3_{Bi}}{|C3_{Bi}| \cdot |Q_{Bi}|} = \frac{0}{1{,}07 \cdot 0{,}67} = 0$$

Aquí concluye el cálculo de la similitud a nivel de bigramas. Obsérvense los valores de similitud de la consulta Q con respecto a las tres categorías. Se aprecia que es entre la consulta Q y la categoría C1 únicamente donde existe similitud; esto es debido a que en el texto de la categoría C1 y de la consulta Q se comparten los bigramas "white-blood" y "blood-cells", mientras que con las categorías C2 y C3 no se comparten bigramas.

En tercer lugar, se prosigue el cálculo de la similitud a nivel de trigramas. Los trigramas se forman a partir de las palabras de los textos de las categorías, tomándolas de tres en tres. Por ejemplo, dado el texto siguiente: "*It is biological system humans produces movement*", se generarían los siguientes trigramas: "*It-is-biological, is-biological-system, biological-system-humans, system-humans-produces, humans-produces-movement*".

En la tabla que aparece en la figura 5.13 puede observarse los cálculos necesarios para conseguir el peso de los trigramas que aparecen en los textos de las categorías C1, C2 y C3.

Puntualizar que también en la figura 5.13 se han omitido los valores iguales a cero para dar mayor claridad a la tabla y proporcionar así una mejor comprensión del caso práctico.

El cálculo de la similitud para los trigramas se realiza de forma similar al cálculo de monogramas y bigramas, esto es, calculando el numerador (ecuación 5.5) y los dos componentes del denominador (ecuaciones 5.6 y 5.7) de la ecuación 5.3 como sigue a continuación:

$$Q_{Tri} \cdot C1_{Tri} = 0{,}48 * 0{,}48 = 0{,}23$$

$$Q_{Tri} \cdot C2_{Tri} = 0$$

$$Q_{Tri} \cdot C3_{Tri} = 0$$

$$|C1_{Tri}| = \sqrt{(0{,}48^2)^1 3} = \sqrt{0{,}23^1 3} = 1{,}72$$

$$|C2_{Tri}| = \sqrt{(0{,}48^2)^9} = \sqrt{0{,}23^9} = 1{,}43$$

$$|C3_{Tri}| = \sqrt{(0{,}48^2)^5} = \sqrt{0{,}23^5} = 1{,}07$$

$$|Q_{Tri}| = \sqrt{0{,}48^2} = \sqrt{0{,}23} = 0{,}48$$

	$tf_{i,j}$							$w_{i,j}=tf_{i,j}*log(D/df_j)$			
Término (j)	$tf_{C1,i}$	$tf_{C2,i}$	$tf_{C3,i}$	$tf_{Q,i}$	df_i	D/df_i	$log(D/df_i)$	$W_{C1,i}$	$W_{C2,i}$	$W_{C3,i}$	$W_{Q,i}$
absorption-various-foods		1			1	3	0,48		0,48		
biological-system-humans			1		1	3	0,48			0,48	
blood-cells-platelets	1				1	3	0,48	0,48			
blood-cells-white	1				1	3	0,48	0,48			
breakdown-absorption-various		1			1	3	0,48		0,48		
cells-red-blood	1				1	3	0,48	0,48			
cells-white-blood	1				1	3	0,48	0,48			
circulating-tissue-composed	1				1	3	0,48	0,48			
composed-fluid-plasma	1				1	3	0,48	0,48			
fluid-plasma-cells	1				1	3	0,48	0,48			
foods-liquids-needed		1			1	3	0,48		0,48		
humans-produces-movement			1		1	3	0,48			0,48	
is-biological-system			1		1	3	0,48			0,48	
is-circulating-tissue	1				1	3	0,48	0,48			
is-responsible-breakdown		1			1	3	0,48		0,48		
lt-is-biological			1		1	3	0,48			0,48	
lt-is-circulating	1				1	3	0,48	0,48			
lt-is-responsible		1			1	3	0,48		0,48		
liquids-needed-sustain		1			1	3	0,48		0,48		
needed-sustain-life		1			1	3	0,48		0,48		
plasma-cells-red	1				1	3	0,48	0,48			
red-blood-cells	1				1	3	0,48	0,48			
responsible-breakdown-absorption		1			1	3	0,48		0,48		
system-humans-produces			1		1	3	0,48			0,48	
tissue-composed-fluid	1				1	3	0,48	0,48			
various-foods-liquids		1			1	3	0,48		0,48		
white-blood-cells	1			1	1	3	0,48	0,48			0,48

Figura 5.13: Representación de los trigramas de los textos según el modelo de espacio vectorial.

Una vez calculados los componentes del numerador y denominador, se calcula el grado de similitud de la consulta Q con respecto a las categorías C1, C2 y C3 como sigue:

$$Sim_{Tri}(C1, Q) = \frac{Q_{Tri} \cdot C1_{Tri}}{|C1_{Tri}| \cdot |Q_{Tri}|} = \frac{0{,}23}{1{,}72 \cdot 0{,}48} = 0{,}27$$

$$Sim_{Tri}(C2, Q) = \frac{Q_{Tri} \cdot C2_{Tri}}{|C2_{Tri}| \cdot |Q_{Tri}|} = \frac{0}{1{,}43 \cdot 0{,}48} = 0$$

$$Sim_{Tri}(C3, Q) = \frac{Q_{Tri} \cdot C3_{Tri}}{|C3_{Tri}| \cdot |Q_{Tri}|} = \frac{0}{1{,}07 \cdot 0{,}48} = 0$$

Aquí concluye el cálculo de la similitud a nivel de trigramas. Obsérvense los valores de similitud de la consulta Q con respecto a las tres categorías. Se

aprecia que únicamente entre la consulta Q y la categoría C1 existe similitud; esto es debido a que en el texto de la categoría C1 y de la consulta Q se comparte sólo el trigrama "white blood cells", mientras que con las categorías C2 y C3 no se comparten trigramas.

Para terminar con el caso práctico, se calcula la similitud total, siguiendo la ecuación 5.4, la cual se obtiene ponderando los valores de similitud obtenidos a nivel de monogramas, bigramas y trigramas.

$$\begin{aligned} Sim_{Global}(C1,Q) &= 0{,}7 \cdot Sim_{Tri}(C1,Q) + 0{,}2 \cdot Sim_{Bi}(C1,Q) \\ &+ 0{,}1 \cdot Sim_{Mono}(C1,Q) = 0{,}7 \cdot 0{,}27 + 0{,}2 \cdot 0{,}55 \\ &+ 0{,}1 \cdot 0{,}58 = 0{,}357 \end{aligned}$$

$$\begin{aligned} Sim_{Global}(C2,Q) &= 0{,}7 \cdot Sim_{Tri}(C2,Q) + 0{,}2 \cdot Sim_{Bi}(C2,Q) \\ &+ 0{,}1 \cdot Sim_{Mono}(C2,Q) = 0{,}7 \cdot 0 + 0{,}2 \cdot 0 \\ &+ 0{,}1 \cdot 0{,}15 = 0{,}015 \end{aligned}$$

$$\begin{aligned} Sim_{Global}(C3,Q) &= 0{,}7 \cdot Sim_{Tri}(C3,Q) + 0{,}2 \cdot Sim_{Bi}(C3,Q) \\ &+ 0{,}1 \cdot Sim_{Mono}(C3,Q) = 0{,}7 \cdot 0 + 0{,}2 \cdot 0 \\ &+ 0{,}1 \cdot 0{,}2 = 0{,}02 \end{aligned}$$

Se puede observar en los resultados anteriores, cómo la similitud mayor corresponde a la calculada entre la consulta Q y la categoría C1. Desde que se realizó el cálculo de la similitud a nivel de monogramas viene siendo así, aunque se aprecia mucha mayor diferencia cuando se realiza el cálculo global de similitud. A continuación se presenta un resumen de los valores de similitud obtenidos en forma de tabla.

Tabla 5.1: Tabla resumen de valores de similitud.

Categoría	$Sim_{Mono}(C_i,Q)$	$Sim_{Bi}(C_i,Q)$	$Sim_{Tri}(C_i,Q)$	$Sim_{Blobal}(C_i,Q)$
C1	0,56	0,55	0,27	0,357
C2	0,15	0	0	0,015
C3	0,2	0	0	0,02

5.9.4. El Proceso Automático

Es necesaria una fase de entrenamiento previa a hacer uso del proceso de anotación automático. En esta etapa, el ingeniero de conocimiento (administrador de FLERSA) define las anotaciones básicas que formarán el Corpus. Se necesita, al menos, una anotación manual para cada concepto de la taxonomía que se quiere entrenar para ser usada en anotaciones automáticas.

Después, técnicas de aprendizaje como las estudiadas en la subsección 5.9.2 pueden usar el texto de cada anotación perteneciente al Corpus para calcular el perfil de un concepto.

Cuando se anota un documento web en modo automático, el sistema FLERSA es capaz de trabajar tanto a nivel global como a nivel local. A nivel global, se considera todo el texto del documento web como un fragmento de texto. A nivel local, el documento es dividido en fragmentos de texto a nivel de párrafo. Después, se lleva cabo un proceso de categorización del texto para cada fragmento. La figura 5.14 ilustra el flujo de datos general para el

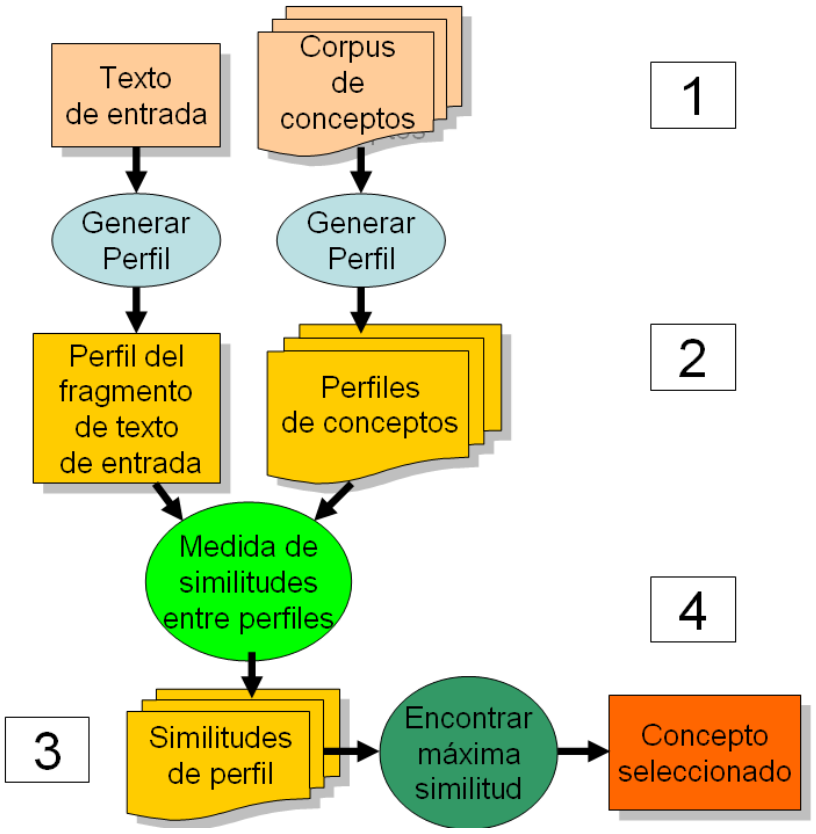


Figura 5.14: Diagrama de flujo para la categorización de texto.

enfoque híbrido de categorización de texto. Las cajas cuadradas representan estados y las ovaladas acciones. Como se puede observar, el proceso automático de anotación semántica consiste en los cuatro pasos que se describen a continuación:

1. Un nuevo texto de entrada -un fragmento de texto o un documento completo- llega al sistema para su clasificación. El sistema tiene, al

menos, una anotación manual para cada concepto con el que trabaja y que forma el Corpus del concepto.

2. Se calcula el perfil de frecuencia de los n-gramas del texto de entrada siguiendo la ecuación 5.2. El sistema ha precomputado previamente los perfiles de cada concepto modelado en las anotaciones del Corpus.

3. El sistema compara el perfil del texto de entrada con respecto a los perfiles precomputados de cada uno de los conceptos modelados. Se usa la medida de la distancia entre perfiles que proporciona la ecuación 5.3 y que se calcula fácilmente. Se calcula la similitud de monogramas, bigramas y trigramas separadamente, de manera que se necesita un valor de similitud ponderado, como en la ecuación 5.4.

4. El sistema clasifica el fragmento de texto como perteneciente al concepto que tenga la similitud más alta. También se considera la posibilidad de no clasificar el texto de entrada cuando su valor de similitud es inferior al de un valor umbral, de manera que no se les asigna categoría a los textos pertenecientes a categorías no entrenadas en el sistema. Cuando un fragmento de texto supera el valor umbral, es correctamente clasificado y se le asigna una anotación semántica similar a las estudiadas en la sección 5.8, en la que se incluye información de categorización (propiedad *Related*).

5.10. Recuperación de Información

La principal ventaja que supone el enriquecimiento con metadatos de los contenidos web es la mejora en la calidad de los resultados que se obtienen cuando se realizan tareas de recuperación de información.

Mediante la explotación de las anotaciones semánticas introducidas, es posible realizar búsquedas "inteligentes" y obtener los resultados esperados. Por ejemplo, en cualquier CMS que contenga artículos sobre coches sería lógico realizar la siguiente búsqueda expresada en lenguaje natural: "*Consumo en coches Ford*", la cuál se traduciría en una búsqueda léxica literal de los términos que aparecen en la consulta y cuyos resultados serían impredecibles; en el caso de que los términos de búsqueda no aparezcan o se utilicen sinónimos de los mismos, la búsqueda no ofrecería resultados. Sin embargo, si realizásemos esta misma búsqueda en un S-CMS como FLERSA, el sistema utilizaría la información semántica de su base de conocimiento para devolver como resultado los vínculos a aquellos artículos que hayan sido catalogados, a nivel global, como que tratan de coches de marca "Ford" y que, a nivel local, contengan párrafos relacionados con el concepto "coste de mantenimiento" (inferido a partir de la palabra "consumo" que aparece en la búsqueda).

Es posible combinar en un S-CMS las técnicas tradicionales de recuperación de información con técnicas basadas en la semántica introducida por las anotaciones. Los tipos de búsqueda que se podrían realizar son los siguientes:

- **Basada en palabras clave**. Se trata del tipo de búsqueda tradicional en la que a partir de unas palabras clave introducidas por el usuario a modo de términos de búsqueda, se presentan como resultados los contenidos web donde éstas se han localizado. Su principal problema es que no ofrecen buenos resultados cuando se usa el lenguaje natural como términos de búsqueda. Por ejemplo, un usuario podría acceder sin problemas a los artículos de un CMS que tratan sobre la marca "Ford" usando esta misma palabra como término búsqueda, pero si la consulta es más elaborada como por ejemplo "artículos que hablan de la seguridad de los coches Ford" es muy probable que la consulta no ofreciera ningún resultado.

- **Basada en las propiedades de las anotaciones**: A través de una consulta, también es posible filtrar artículos basándose en cualquier propiedad de las anotaciones semánticas definidas en ellos. En el caso de utilizar la ontología FLERSA-Ontology, sería posible recuperar artículos a partir de propiedades como: Página de anotación, autor, cuerpo de anotación, contexto, fecha de creación, granularidad, fecha de modificación, conceptos que se tratan en las anotaciones, sección y tipo. Por ejemplo, en el sistema FLERSA se podrían buscar los artículos anotados este último mes, o los que hablan del coche marca "Audi" que fueron anotados por el usuario "José Luis".

- **Basada en conceptos**: Es ella se combinarían las dos búsquedas anteriores. La idea es vincular una lista de palabras clave con los conceptos de una taxonomía. Cuando el usuario introduzca los términos de búsqueda, el sistema determinará automáticamente los conceptos que están asociados a ellos y ofrecerá como resultados los contenidos web en cuyas anotaciones semánticas figuren los conceptos que son objeto de búsqueda. En este tipo de búsqueda es posible asociar al mismo concepto tanto términos equivalentes como sinónimos de forma que se mejora la búsqueda tradicional. También se permite inferir conceptos específicos a partir de otros más generales, ofreciendo así mejores resultados. Por ejemplo, un usuario podría realizar anotaciones semánticas en páginas que tratan sobre coches para especificar el modelo y marca de cada uno; después, el usuario podría hacer una consulta buscando artículos que tratan de conceptos generales como *"coche"* o *"audi"* y obtener resultados específicos como son artículos que tratan de *"bmw"* o de *"A4"*. Destacar que se usan ontologías a modo de taxonomías para esta tarea; cuando se realiza una consulta, se consideran los conceptos

que concuerdan con los términos de la consulta y sus descendientes; el resultado de la búsqueda se compone de la lista de artículos que tratan sobre esos conceptos.

- **Consultas en lenguaje natural**. Es el tipo de búsqueda más acorde con la filosofía de Web Semántica. El usuario realiza una consulta expresada en lenguaje natural y el sistema la analiza, siguiendo el enfoque híbrido estudiado en el apartado 6.6.8, para determinar los conceptos objeto de búsqueda. Por ejemplo, ante la consulta "artículos que tratan sobre el consumo de los coches Ford" el S-CMS inferirá los conceptos "Gastos de mantenimiento" y "Ford" y devolverá como resultado los artículos en los que intervienen ambos.

  Queda claro que los tipos de búsqueda explicados anteriormente presentan muchas ventajas, aunque sólo se usen para realizar búsqueda contextual de información contenida en un CMS particular.

5.11. Conclusiones

En este capítulo se han presentado todos los aspectos que se consideran necesarios a la hora transformar los CMS en sus equivalentes semánticos y que se han llevado a la práctica en el sistema FLERSA. El capítulo comienza con un estudio de los requisitos de diseño que deben cumplir este tipo de sistemas, continuando con una propuesta de arquitectura específica que permita adaptar la existente a las nuevas características de éstos, y terminando con un estudio detallado de las principales facetas que realizan: Las técnicas de anotación semántica, los procesos de anotación tanto manual como automatizada, y los procesos de recuperación de información basada en información semántica.

Cabe destacar que la principal característica que proporciona FLERSA, y que debería aportar cualquier S-CMS, es su funcionalidad para trabajar con anotaciones semánticas realizadas sobre contenido web. Éstas se basan en una ontología base inspirada en el framework Annotea, cuyo propósito es dar soporte a la definición de anotaciones de infraestructura; a partir de éstas se pueden definir anotaciones adicionales, usando conceptos y propiedades de otras ontologías. Para las anotaciones manuales se ha usado una técnica innovadora de marcado de rangos de texto flexible, basada en el estándar RDFa, para soportar la evolución de los documentos web anotados de forma más efectiva que XPointer. Para las anotaciones automáticas se han usado técnicas híbridas de aprendizaje (Modelo de Espacio Vectorial y n-gramas) para determinar los conceptos de los que se trata en el contenido de los documentos web, basándose en anotaciones previas usadas a modo de Corpus.

Capítulo 6

COM_SEMANTIC: El Componente para Joomla

Vive como si fueras a morir mañana y
aprende como si fueras a vivir siempre.

Gandhi

Resumen: El resultado de todo el trabajo desarrollado para la tesis se materializa en una extensión de Joomla, un conocido Sistema de Gestión de Contenidos. El componente se llama *com_semantic* y está disponible gratuitamente para su descarga en la dirección web `http://salmer.sourceforge.net`. Ha sido lanzado bajo licencia GNU/GPL Affero versión 3 y proporciona una implementación de un sistema de anotación semántica, tanto manual como automatizada. En el presente capítulo se describen los detalles de desarrollo de la herramienta, así como las funcionalidades que proporciona.

6.1. Introducción

Sabemos que es posible extender la funcionalidad que ofrecen los CMS mediante la agregación de módulos o plugins. La idea principal que se propone es trabajar en esta línea, desarrollando un módulo o extensión que haga uso de APIs, librerías y/o middleware a modo de infraestructura, ya que éstas proporcionan primitivas básicas para trabajar con las tecnologías de la Web Semántica a partir de las cuales componer funciones complejas desde un nivel de abstracción superior.

Una de las primeras tareas con las que se pretende enriquecer a los CMS es la de ofrecer contenidos que incorporen información semántica. Esta información semántica se ofrecerá conjuntamente con los contenidos en forma metadatos, los cuales serán descritos utilizando un lenguaje de marcas recomendado para ello por la W3C, como por ejemplo RDFa o RDF. Esta tarea supone una primera aproximación a la publicación en la Web Semántica de los contenidos gestionados por un CMS.

El proceso de enriquecimiento de los contenidos de los CMS con información semántica no consiste en transformar estos contenidos, almacenados en tablas de bases de datos, a RDF/XML. El proceso es más complicado, necesita de vocabularios RDF para representar un determinado ámbito de contenidos, a un nivel de detalle y agregación adecuados. Las anotaciones semánticas han de contar con las propiedades necesarias para describir el recurso que anotan, en nuestro caso fragmentos de texto, algunas de las cuales se podrán mapear sobre estos vocabularios RDF para expresar el tipo de información que se almacena.

En el camino hacia la incorporación de los CMS en la Web Semántica se pasa inevitablemente por la instalación y configuración del software (componentes, librerías, módulos) de un modo sencillo, para acelerar la productividad de los sistemas y comenzar rápidamente a publicar información semántica. Para ellos se precisan soluciones del tipo "instalar y listo" esenciales para simplificar la complejidad que supone el uso de software heterogéneo evitando así perdernos dentro del laberinto de extensiones y módulos que actualmente ofrecen los CMS.

6.2. Objetivos

El objetivo principal es el desarrollo de un componente sobre un CMS que posibilite la extensión de las características del mismo y su transformación en un S-CMS. La instalación del componente sobre un CMS le aporta nuevas funcionalidades tales como la anotación semántica de contenidos tanto manual como automática, así como búsquedas de información mejorada.

En la consecución de este objetivo se ven involucrados diversos procesos encargados de las facetas semánticas con las que pretenden enriquecer los CMS, y que se ocupan, entre otras cosas de:

- La generación de anotaciones semánticas en un lenguaje de marcas adecuado.
- La gestión de la base de conocimiento y los servicios de almacenamiento.

- La recuperación y explotación de la información semántica.

6.3. Requisitos

Durante el desarrollo del componente se han tenido presentes los requisitos para el desarrollo de sistemas de anotación semántica, estudiados en la sección 5.3 del capítulo 5, y que aquí se concretan como sigue a continuación:

- **Requerimiento 1 - formatos estándar.** El componente deberá hacer uso exclusivo de estándares abiertos recomendados por el W3C con objeto de promover la interoperabilidad y la extensibilidad.

- **Requerimiento 2 - diseño colaborativo/centrado en el usuario.** La capacidad del componente para ofrecer trabajo colaborativo está subyugada a la posesión de esta característica por parte del CMS subyacente, esto es, las distintas herramientas que proporcione el componente podrán ser usadas de forma colaborativa en la medida en la que el CMS huésped proporcione esta característica. En cuanto al otro requisito, el componente ofrecerá herramientas centradas en el usuario, las cuales estarán completamente integradas con los navegadores web de tal forma que los usuarios únicamente tengan que interactuar por medio del ratón con el texto del documento donde se hacen las anotaciones y a través de los menús que la herramienta proporcione para ello.

- **Requerimiento 3 - soporte de ontologías.** El uso de ontologías debe ser una pieza clave a la hora de desarrollar el componente, ya que juegan un papel fundamental en los aspectos relacionados con la representación del conocimiento (FLERSA-ontology) e integración con otras herramientas de la Web Semántica.

- **Requerimiento 4 - soporte para formatos de documentos heterogéneos.** Los documentos con los que trabajan los CMS son páginas web. Además, éstos proporcionan sus contenidos en forma de artículos. Dado que éstos pueden contener cantidades variables de texto y distinto contenido multimedia (imágenes, aplicaciones flash o java), la herramienta de anotación que proporcione el componente deberá ser capaz de salvar estas heterogeneidades y de trabajar con los distintos componentes que aparezcan en un documento web. En cuanto al texto que compone los artículos, habrá que considerar que su longitud es variable y que el contenido de éstos puede tratar sobre diferentes temas, por lo que se hace se hace necesario delimitar de alguna manera, dentro

de los contenidos, los fragmentos de texto que tratan sobre un determinado concepto. También es necesario generar información semántica en la que se describa el contenido de estos fragmentos de texto marcados.

- **Requerimiento 5 - evolución de documentos.** Las anotaciones semánticas realizadas sobre un artículo del CMS deben permanecer consistentes frente a revisiones, borrados o movimientos que afecten al texto que compone el artículo, es decir, se deben mantener correctamente las anotaciones conforme cambian los documentos web.

- **Requerimiento 6 - almacenamiento de anotaciones.** Las anotaciones semánticas que se realicen sobre un artículo del CMS se almacenarán como parte del documento web preferible ya que ayuda a mantener las anotaciones consistentes frente a nuevas versiones del documento. Se almacenarán en el repositorio de contenidos para facilitar futuras labores de búsqueda e inferencia de información.

- **Requerimiento 7 - automatización.** El componente proporcionará una herramienta de anotación de contenido web que permita tanto anotaciones manuales como automatizadas. La anotación semántica automática deberá usar mecanismos que permitan determinar los conceptos de los que trata el contenido web.

- **Requisito 8 - integración.** No se pretende desarrollar toda la infraestructura del sistema de anotación semántica. Se usará un CMS como infraestructura subyacente, sobre el que se extenderá su funcionalidad mediante la instalación de un componente semántico. El componente será el resultado de la integración de distintos módulos existentes, los cuales proporcionarán parte de las facilidades de programación y/o comunicación, junto con nuevos módulos que se desarrollarán desde cero para componer las funcionalidades que se fijaron en los objetivos.

- **Requisito 9 - interfaz de usuario web.** El ámbito de trabajo natural de los CMS es el navegador Web así, su variante semántica debe ser capaz de trabajar y poder administrarse desde el mismo entorno. También se hace necesaria la utilización de lenguajes de programación en el lado cliente que posean la potencia adecuada y cuyo uso esté estandarizado por parte de los navegadores.

- **Requisito 10 - compatibilidad multi-navegador.** Hoy día existen gran variedad de navegadores web, cada cuál con sus peculiaridades. Se buscará dentro de lo posible la compatibilidad de la herramienta al menos con los navegadores más difundidos.

Además de lo anterior, el componente proporcionará una herramienta de búsqueda capaz de trabajar de diferentes formas, permitiendo desde una

búsqueda clásica, hasta otros tipos de búsquedas semánticas en las que se utilicen metadatos y ontologías para mejorar las búsquedas.

La instalación de la herramienta de anotación será fácil y desde entorno web; estará completamente integrada con el navegador web de tal forma que los usuarios únicamente tengan que interactuar con él para conseguir su puesta en marcha.

6.4. Características

El componente se llama **"com_semantic"** porque es un "**com**ponente **semántic**o" encargado de aportar a los CMS las características que necesitan para que puedan transformarse en sus equivalentes semánticos, los S-CMS.

El componente lo forman dos grandes partes: Una de administración de la herramienta o *back-end* y otra accesible a los usuarios o *front-end*. A su vez, dentro del front-end el usuario puede encontrar dos grandes bloques: Uno llamado *"FLERSA Annotation Tool"* dedicado a una herramienta de anotación semántica de contenidos y otro llamado *"Semantic-powered Search"* dedicado a la búsqueda de información dentro del CMS basada en diferentes criterios.

La herramienta de anotación "FLERSA Annotation Tool" es una herramienta de marcado semántico diseñada para generar anotaciones semánticas en el contenido de documentos web una vez que éstos han sido creados. El autor del documento será el usuario que cree el marcado; los demás usuarios se beneficiarán de la explotación del conocimiento asociado. Se ha desarrollado según la arquitectura cliente-servidor, lo que posibilita que múltiples usuarios realicen anotaciones de forma simultánea (centrada en el usuario) y lo que es más importante que puedan colaborar entre ellos y reusar documentos inteligentes (Handschuh et al., 2003a). Utiliza la técnica de marcado de rangos flexibles de texto estudiada en la sección 5.7 del capítulo 5. Ésta está basada en el estándar RDFa, la cual permite conseguir la evolución de los documentos anotados de forma más efectiva que XPointer.

Como quedó justificado en el capítulo 5, RDFa es el candidato ideal para ser usado como lenguaje de anotación debido a que puede ser incrustado directamente en contenidos y a que, a diferencia de RDF, no repite contenido cuando es usado sobre datos estructurados como es el caso de XHTML. Su funcionamiento es muy similar al de los microformatos, aunque utiliza RDF como modelo para describir un elemento de información del contenido. Además, por medio de analizadores sintácticos, es posible obtener los datos RDF a partir del código fuente en XHTML.

Para la anotación semántica automática, la herramienta de anotación usa

el enfoque híbrido basado en técnicas de aprendizaje tales como el modelo de espacio vectorial combinado con n-gramas, estudiado en la sección 5.9.2, el cual permite determinar los conceptos de los que trata el contenido web.

La herramienta de búsqueda "Semantic-powered Search" es capaz de realizar diferentes tipos de búsquedas: Clásica, basada en palabras clave; guiada por la información recogida en las anotaciones y, la más interesante, guiada por los conceptos que se tratan en el contenido web. En este último tipo de búsqueda se hace uso de ontologías a modo de taxonomías de conceptos y posibilitan la inferencia de información necesaria para mejorar las búsquedas.

El componente hace uso exclusivo de estándares abiertos tales como XML, RDF, RDFa y OWL con objeto de promover la interoperabilidad y la extensibilidad.

En la implementación del componente se ha buscado, en la medida de lo posible, la compatibilidad multi-navegador (cross-browser). No todos los navegadores soportan la especificación W3C DOM Range, por lo tanto, sería deseable conseguir la compatibilidad multi-navegador desde la implementación. Se ha conseguido la compatibilidad con los navegadores más difundidos como son: Internet Explorer, Mozilla Firefox, Chrome y Opera.

Se trata de un componente que posee un nivel de acoplamiento débil, es decir, la implementación que se ha realizado puede ser adaptada fácilmente a otro CMS. Los servicios que proporciona el lado servidor se han integrado a la infraestructura Web subyacente cuyos detalles técnicos se presentarán más adelante en la sección 6.5.

La principal ontología que usa el componente es FLERSA-ontology, estudiada en la sección 5.6. Esta ontología se usa como estructura base de anotación para cualquier anotación semántica en un documento web, de manera que se crea una instancia para cada anotación que se hace. La ontología subyacente también permite la posibilidad de usar vocabularios (microformatos) alternativos cuando se hace una anotación.

Otra de las características principales de FLERSA es el almacenamiento dual de anotaciones semánticas, esto es, se almacenan en la base de datos del lado servidor en lenguaje de definición de metadatos RDF y, por otro lado, se almacenan incrustadas en el mismo documento donde se anota en lenguaje RDFa de forma totalmente transparente al usuario. Esta característica une las ventajas del modelo de almacenamiento de anotaciones centralizado a las del incrustado: Inferencia de nuevo conocimiento a partir de la base de datos de anotaciones, disponibilidad de anotaciones autocontenidas en el propio documento, el libre acceso a los metadatos de los documentos web por parte de indexadores, buscadores y otros tipos de servicios semánticos para mejorar las búsquedas y, por último, la posibilidad de proporcionar información sobre

la estructura interna de los documentos, así como la relación entre ellos.

Las principales funcionalidades que aporta la herramienta son:

- Creación de anotaciones asociadas a un rango de texto.
- Edición de anotaciones preexistentes
- Borrado de una anotación preexistente.
- Borrado de todas las anotaciones del documento.
- Almacenamiento permanente de las anotaciones.
- Creación de anotaciones globales a la página (Global).
- Visualización del RDF generado en la página (W3C's RDFa Distiller).
- Búsqueda inteligente de anotaciones en base a las propiedades que se han anotado. Se realiza inferencia en las taxonomías de conceptos cuando la búsqueda se realiza sobre la propiedad *related*.

6.5. Detalles del Desarrollo

En esta sección se explican los detalles de desarrollo del componente *"com_semantic"* desde un punto de vista técnico: Desde los componentes del sistema utilizados hasta las estrategias y utilidades usadas durante su codificación.

La figura 6.1 muestra los detalles de implementación de la arquitectura mostrada anteriormente en la figura 5.1 del capítulo 5. En ella se representa la arquitectura del S-CMS, especificando con detalle los componentes que se han usado en su consecución. A continuación se justifica porqué se han usado estos componentes y no otros, así como la utilidad que proporcionan al sistema.

En cuanto a la elección de un candidato de sistema CMS donde realizar el desarrollo de las soluciones que se definieron en el capítulo 5, hay que considerar que los CMS basados en PHP son los más utilizados actualmente y que ofrecen una mayor flexibilidad y capacidad de personalización. Se pretende evitar la dispersión tecnológica que aparece en algunos sistemas de este tipo y trabajar hacia la integración completa en la Web Semántica. Es por estas razones, junto con el estudio realizado en el capítulo 2, por lo que se ha seleccionado Joomla como el CMS sobre el que realizar el desarrollo. Además, Joomla es una herramienta de código libre bajo licencia GNU/GPL que está

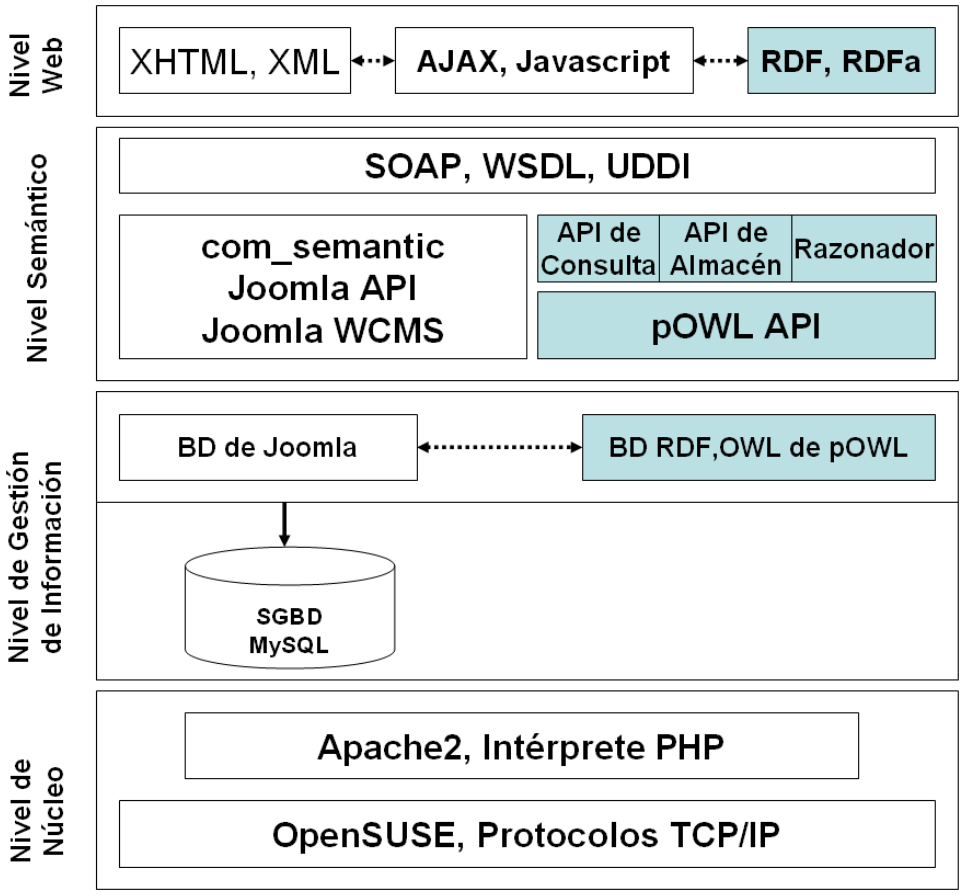


Figura 6.1: Detalles de implementación de la arquitectura.

programada de forma modular y nos permite incluir extensiones con la funcionalidad que se necesite. En particular, se ha implementado el componente com_semantic para Joomla, el cual aporta la herramienta de marcado de anotaciones web "FLERSA Annotation Tool", la herramienta de búsqueda semántica "Semantic-powered Search" y la herramienta de administración del componente "Semantic Annotation Configuration".

Como framework para trabajar con ontologías se ha usado pOWL[1]. Se trata de un marco de trabajo Opensource para el análisis sintáctico, almacenamiento, consulta, manipulación, servicio y serialización de bases de conocimiento en un entorno colaborativo Web. El entorno pOWL (Auer, 2005) proporciona a FLERSA soporte multi-ontología, facilitando la creación de una base de conocimiento compuesta por vocabularios consensuados y taxonomías a partir de las cuales llevar a cabo anotaciones semánticas.

[1] http://sourceforge.net/projects/powl/

En cuanto a la programación de la herramienta cabe destacar que se han utilizado diversas técnicas de programación como es el caso del patrón de diseño MVC y la tecnología AJAX (*Asincronous Javascript And XML*, Javascript y XML asíncrono). A la hora de programar en el lado servidor ha habido que adecuarse al lenguaje anfitrión en el que está desarrollado Joomla, por tanto, se ha programado en PHP haciendo uso de las APIs que ofrecen Joomla y pOWL.

En cuanto al lado cliente, se ha usado el lenguaje de programación Javascript ya que es un lenguaje compatible con todos los navegadores actuales; además se ha usado en lo posible la tecnología AJAX, proporcionada por la librería Javascript Mootools[2], la cual agiliza la interacción entre el usuario y la Web mediante la transferencia asíncrona de mensajes de datos en formato XML entre servidor y cliente.

Se ha recurrido a la programación de una librería Javascript específica para solucionar el problema de compatibilidad multi-navegador. Se han implementado las funciones que permiten obtener la funcionalidad fijada en la especificación DOM Range Level 2, independientemente del navegador que se use. Por tanto, las herramientas desarrolladas son compatibles con los navegadores más difundidos, como son: Internet Explorer, Mozilla Firefox, Chrome y Opera.

A la hora de implementar el proceso de definición de rangos flexibles de anotación explicado en la sección 5.7, se ha recurrido a la librería Javascript anterior y al uso de una hoja de estilos CSS para la delimitación visual de los rangos que se definan. La generación de RDFa se lleva a cabo de forma automática conforme se realiza el marcado de la anotación semántica. La automatización se ha programado en Javascript y se encarga de añadir etiquetas *"SPAN"* con metadatos acerca de los textos a los que se les realiza la anotación semántica; en esta tarea es esencial el uso de la función *"surround"*, que proporciona la tecnología DOM Range Level 2, para delimitar el fragmento de texto que se está marcando para ser anotado. Algunos de los metadatos que actualmente se están guardando son: Autor de la anotación, contexto donde se anota (URI), fecha de creación, texto al que se refiere, granularidad del texto (letra, palabra, frase, párrafo o texto libre) y tipo de anotación (texto o imagen).

Los principales lenguajes usados para el desarrollo del componente son PHP y Javascript. Las sentencias SQL que se incluyen junto al componente son usadas para la creación de infraestructura de base de datos y ejemplos. También se ha elaborado un fichero en formato XML encargado de la configuración de la instalación del componente. El componente lo forman 74476 líneas de código fuente, de las cuales el 12 % fueron desarrolladas desde cero

[2]http://mootools.net/

y el 88 % restante fueron integradas a partir de otros proyectos de código libre.

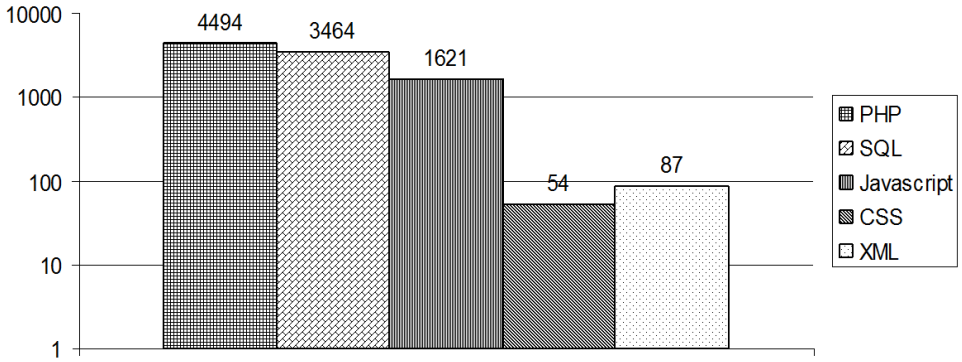


Figura 6.2: Distribución del código fuente desarrollado.

En la figura 6.2 se puede observar la distribución en lenguajes de programación de las líneas de código desarrolladas.

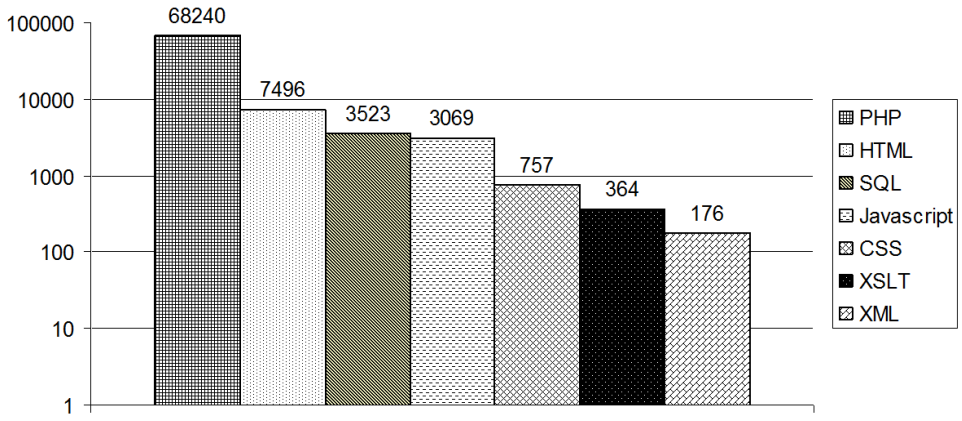


Figura 6.3: Distribución del código fuente de terceros.

En la figura 6.3, se puede observar la misma distribución para el código fuente de terceros que ha sido usado a la hora de desarrollar el componente.

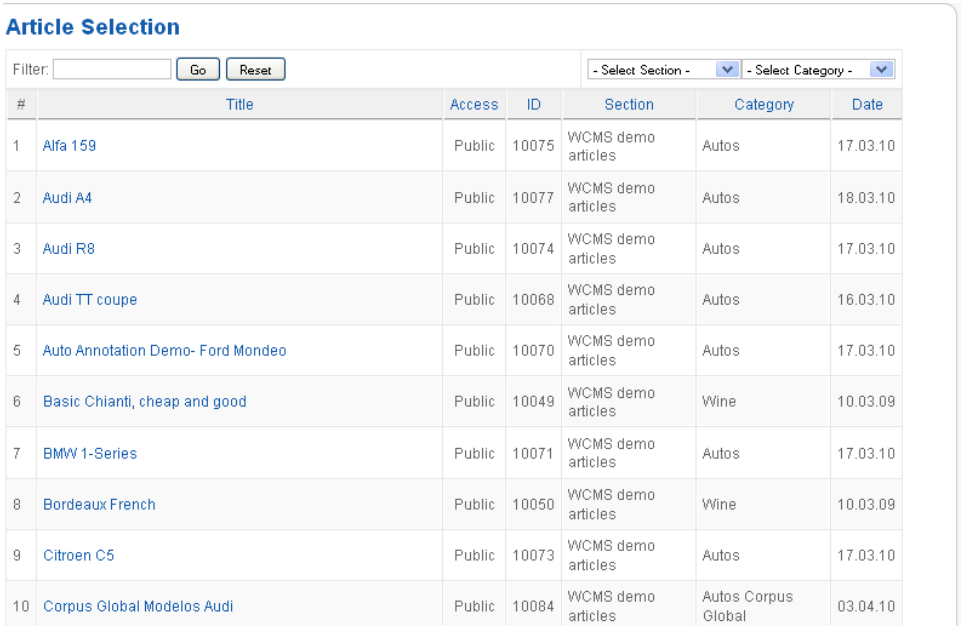
6.6. Funcionalidad de Usuario o Front-End

El término **front-end** en el contexto de los CMS se refiere al área específicamente diseñada para ser accesible a los usuarios y navegadores Web, donde se obtiene acceso a la funcionalidad del sistema.

Figura 6.4: Menú principal del usuario.

En la figura 6.4 se muestra el menú inicial que aparece cuando un usuario entra en un sistema Joomla donde se ha instalado el componente "com_semantic". Las opciones disponibles son:

- **FLERSA Annotation Tool:** Es la opción principal. Desde aquí, los usuarios puede acceder sólo a sus propios artículos (véase figura 6.5).



#	Title	Access	ID	Section	Category	Date
1	Alfa 159	Public	10075	WCMS demo articles	Autos	17.03.10
2	Audi A4	Public	10077	WCMS demo articles	Autos	18.03.10
3	Audi R8	Public	10074	WCMS demo articles	Autos	17.03.10
4	Audi TT coupe	Public	10068	WCMS demo articles	Autos	16.03.10
5	Auto Annotation Demo- Ford Mondeo	Public	10070	WCMS demo articles	Autos	17.03.10
6	Basic Chianti, cheap and good	Public	10049	WCMS demo articles	Wine	10.03.09
7	BMW 1-Series	Public	10071	WCMS demo articles	Autos	17.03.10
8	Bordeaux French	Public	10050	WCMS demo articles	Wine	10.03.09
9	Citroen C5	Public	10073	WCMS demo articles	Autos	17.03.10
10	Corpus Global Modelos Audi	Public	10084	WCMS demo articles	Autos Corpus Global	03.04.10

Figura 6.5: Selección de artículos de usuario.

Una vez que un artículo ha sido seleccionado aparece un panel con la barra de herramientas de anotación como el de la figura 6.6. Éste proporciona diferentes opciones para trabajar con las anotaciones semánticas asociadas al artículo seleccionado. En los siguientes apartados se va a ha comentar la funcionalidad que ofrece.

Figura 6.6: Panel de barra de herramientas de FLERSA.

- **Semantic-powered Search:** Desde esta opción, los usuarios pueden hacer consultas sobre los artículos del CMS; beneficiándose de las ventajas que ofrecen los metadatos incorporados en las anotaciones semánticas.

6.6.1. Creación de Anotaciones Manuales

La herramienta "*FLERSA Annnotation Tool*" permite la creación de nuevas anotaciones semánticas de forma manual. El objetivo del proceso de anotación es asociar metadatos a un fragmento de texto o a una imagen.

El proceso consiste en los siguientes pasos: En primer lugar tiene que seleccionarse un fragmento de texto o imagen mediante el ratón y después pulsar sobre el botón **"New"** de la barra de herramientas de *"FLERSA Annotation Tool"*; entonces, la barra de herramientas se transformará en un editor de anotaciones como el mostrado en la figura 6.7. En este punto, se asigna un identificador único al fragmento de texto o imagen seleccionado, este valor identifica a la anotación semántica y es siempre usado como sujeto en las sentencias RDF (tripletas sujeto-predicado-objeto) que determinan la información semántica de la anotación. Cada anotación se considera como una instancia de una ontología de infraestructura, de manera que se necesita una sentencia RDF para cada propiedad de la anotación. Las propiedades por defecto (también llamadas predicados) que componen una anotación básica son: *annotates, author, body, context, created, modified, granularity and section.*

Las propiedades anteriores son generadas automáticamente por la herra-

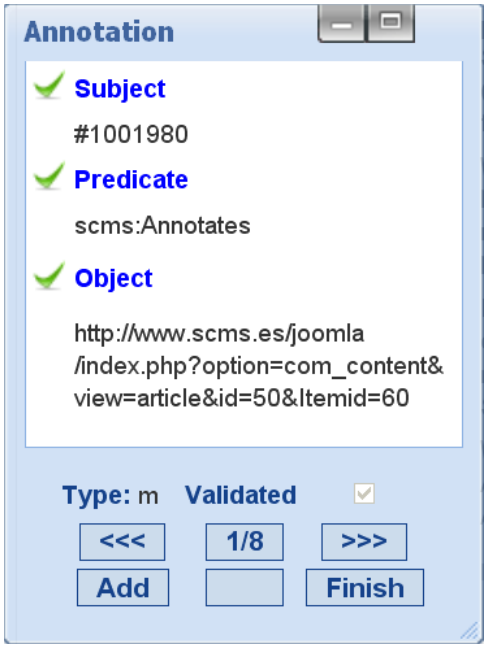


Figura 6.7: Ventana de inspección.

mienta de anotación de acuerdo con el fragmento de texto o imagen que está siendo anotada. También es posible añadir manualmente las siguientes propiedades que pertenecen a la ontología de infraestructura:

- **Related**: Estable una relación entre la anotación y las ontologías de referencia que se usan a modo de taxonomías. Se usa para asociar una anotación con el concepto que se trata en ella. Es una de las propiedades más importantes porque se usa para especificar el concepto del que trata la anotación; en el futuro será muy útil para mejorar las búsquedas con ayuda de la base de datos de anotaciones.
- **Type**: Indica el tipo de anotación que se hace. Los tipos de anotación son: *example, advice, change, seealso, explanation, question and comment*. Estos tipos han sido heredados del marco de trabajo Annotea.

Volviendo a la figura 6.7, destacar que las sentencias RDF asociadas a una anotación pueden ser consultadas individualmente en la ventana de inspección mediante los botones de flecha hacia la izquierda y derecha. El botón **"Add"** se usa para añadir propiedades a la anotación actual. El botón **"Finish"** se usa para terminar la anotación manual actual. La etiqueta **"Type"** indica el valor "`m`" cuando la anotación se ha realizado de forma manual, y el valor "`a`" cuando se ha realizado de forma automatizada. La etiqueta

"Validated" se usa únicamente con anotaciones automáticas, sirve para que el usuario indique si el concepto inferido en la anotación, a partir del marcado de texto que se anota, es válido. Una vez validada una anotación, el fragmento de texto asociado a la misma pasa a formar parte del Corpus de entrenamiento del concepto que se anota en ella.

Cuando se añade una nueva propiedad a una anotación, la ventana de inspección mostrada en la figura 6.7 cambia a la ventana de adicción mostrada en la figura 6.8.

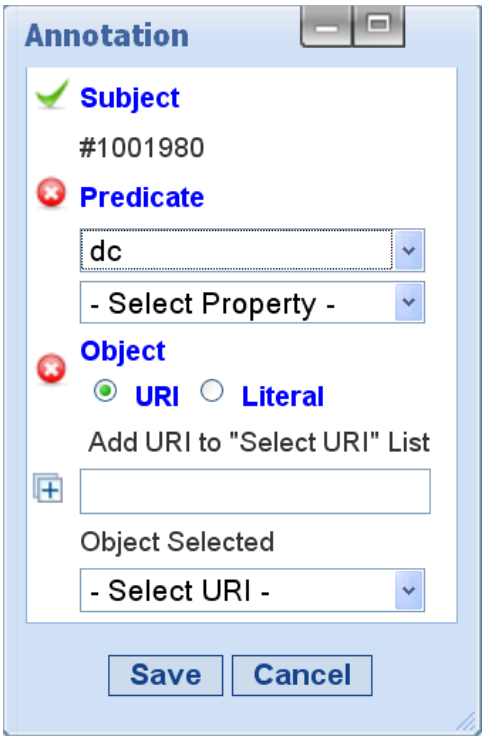


Figura 6.8: Ventana de adicción.

La herramienta de anotación es capaz de manejar diferentes vocabularios (microformatos) que aparecerán en el campo **"Predicate"**. El entorno de pruebas de la herramienta está configurado para usar FOAF, RDF(S) y Dublín Core, así como el vocabulario de infraestructura (llamado *"scms"*) para la definición de las propiedades de las anotaciones. De acuerdo con el estándar RDF, sólo se puede usar como valor del campo **"object"** de una sentencia RDF una URI (Universal Resource Identifier) o un literal. Las URIs pueden introducirse manualmente o a través de la ventana que se muestra en la figura 6.9, llamada **"Ontology Selector"**.

La principal función del selector de ontologías es mostrar las ontologías

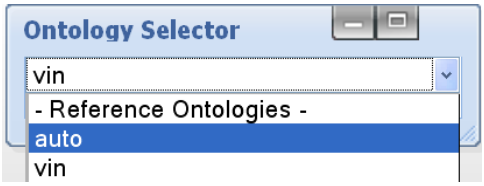


Figura 6.9: Selector de ontologías.

disponibles que ofrece la herramienta FLERSA, que pueden ser usadas en sentencias RDF.

La figura 6.10 muestra un ejemplo de ontología sobre coches cuyas clases son usadas a modo de taxonomía de conceptos. Cuando se añaden sentencias RDF a una anotación, los conceptos son usados en el campo **"Object"** para establecer relaciones que describen qué tema se considera asociado al fragmento de texto o imagen asociada (propiedad *"Related"*).

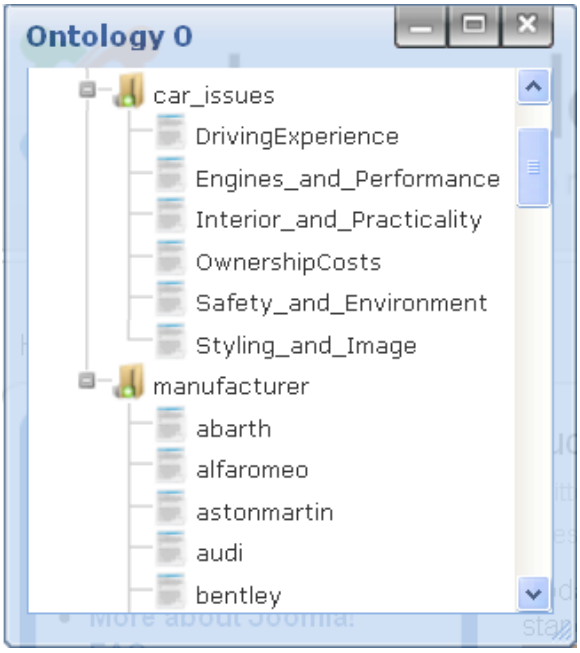


Figura 6.10: Taxonomía de coches.

6.6.2. Edición de Anotaciones.

La herramienta de anotación permite la edición de anotaciones preexistentes. Los pasos son: Pulsar sobre el botón **"Edit"** de la barra de herramientas de la figura 6.6 y después pulsar sobre una anotación existente. Se podrá observar cómo el icono del ratón cambia de forma desde punta de flecha a cruz. Finalmente, la barra de herramientas se transforma en la ventana de inspección de la figura 6.7 desde la cual fue creada de forma manual previamente.

6.6.3. Borrado de Anotaciones.

La herramienta también soporta el borrado de anotaciones existentes. Los pasos son: Pulsar sobre el botón **"Delete"** de la barra de herramientas y después pulsar sobre una anotación existente. Se puede observar cómo la delimitación coloreada de la anotación desaparece.

6.6.4. Borrado de Todas las Anotaciones.

Esta función es muy útil cuando se pretende borrar cada una de las anotaciones de un artículo del CMS. Se consigue pulsando el botón **"Reset"** de la barra de herramientas.

6.6.5. Almacenamiento Permanente de Anotaciones.

Cuando se pulsa en el botón **"Save"** del panel de la barra de herramientas, todas las anotaciones semánticas de la página web que esté activa son guardadas.

Las anotaciones son almacenadas en la base de datos del servidor en el lenguaje de definición de metadatos RDF. Cuando un usuario pulsa en el botón *"Save"*, las anotaciones son codificadas en formato XML y son enviadas al servidor usando tecnología AJAX para ello.

Por otra parte, las anotaciones son almacenadas incrustadas en el mismo documento donde son realizadas en lenguaje RDFa y de una forma totalmente transparente al usuario. Se usa tecnología Javascript para incrustar automáticamente el código RDFa de las anotaciones dentro del documento.

6.6.6. Creación de Anotaciones Globales.

Algunas veces se necesitan anotaciones donde su ámbito es una página completa. Los usuarios pueden estar interesados en establecer a través de una anotación semántica el autor de una página web o de especificar el concepto que se trata en ella.

El botón **"Global"** de la barra de herramientas de FLERSA lleva a cabo esta funcionalidad. Cuando pulsamos sobre él, aparece la ventana de inspección de la figura 6.6 mostrando información relacionada con la página completa actual. También ofrece la posibilidad de añadir/borrar sentencias RDF.

6.6.7. Visualización de RDF.

El W3C tiene un servicio llamado *RDFa Distiller* para identificar y listar las sentencias RDF de una página web que internamente contiene anotaciones en formato RDFa. Cuando se pulsa sobre el botón **"W3C's RDFa Distiller"** de la barra de herramientas, este servicio es invocado y se muestra una ventana (véase figura 6.11) que contiene el RDF que corresponde a las anotaciones semánticas de la página Web actual (artículo actual).

Figura 6.11: Extracción de RDF a partir de RDFa incrustado.

6.6.8. Generación Automática de Anotaciones Semánticas.

Gracias a la creación de anotaciones semánticas automatizadas, los ingenieros de conocimiento son capaces de entrenar un sistema para que establezca relaciones automáticamente entre anotaciones semánticas dentro de un documento y conceptos proporcionados por taxonomías de la base de conocimiento.

Cuando se está anotando un documento web en modo automático, la herramienta de anotación es capaz de trabajar tanto a nivel global como a nivel local.

Cuando se pulsa el botón **"Local Auto-Annotation"** del panel de la barra de herramientas, se trabaja a nivel local. El documento web es dividido en fragmentos de texto a nivel de párrafo y es marcado con una anotación semántica. Se crea una anotación semántica por párrafo, en la que se incluye la información de categorización de la misma en la propiedad *"Related"*.

Cuando se pulsa el botón **"Global Auto-Annotation"** en el panel de la barra de herramientas, se trabaja a nivel global. Se considera el texto completo del documento web en el proceso de categorización. Se crea una anotación semántica para el texto completo, en la que también se incluye la información de categorización de la misma en la propiedad *"Related"*.

Las sentencias generadas durante el proceso de anotación automatizado pueden ser consultadas desde la ventana de inspección que proporciona la opción **"Edit"** del panel de la barra de herramientas de FLERSA.

6.6.9. Herramienta de Búsqueda Semántica.

Una herramienta de búsqueda avanzada aparece cuando se pulsa en el enlace del menú de usuario llamado **"Semantic-powered Search"** (figura 6.4) similar al que muestra la figura 6.12.

La principal ventaja que supone el enriquecimiento con metadatos de los contenidos web es la mejora en la calidad de los resultados que se obtienen cuando se realizan tareas de recuperación de información.

En particular, en la herramienta *"Semantic-powered Search"* se combinan las técnicas tradicionales de recuperación de información con técnicas basadas en la semántica introducida por las anotaciones. Los tipos de búsqueda que permite realizar son los siguientes:

- **Basada en palabras clave**. Se trata del tipo de búsqueda tradicional en la que, a partir de unas palabras clave introducidas por el usuario

a modo de términos de búsqueda, se presentan como resultados los contenidos web donde se éstos se han localizado (véase figura 6.12).

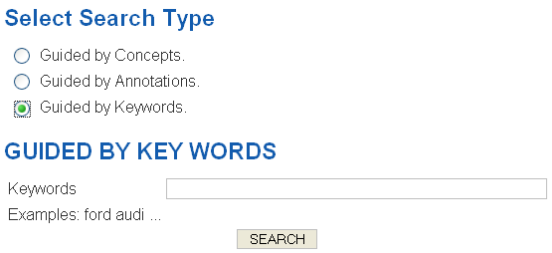


Figura 6.12: Formulario de consulta basada en palabras clave.

- **Basada en las propiedades de las anotaciones**: Se encarga de filtrar artículos en base a las propiedades de las anotaciones semánticas definidas usando FLERSA-ontology (véase figura 6.13). Son las siguientes: Página de anotación, autor, cuerpo de anotación, contexto, fecha de creación, granularidad, fecha de modificación, conceptos que se tratan en las anotaciones, sección y tipo. La propiedad *"Related"* es la más importante puesto que en ella se especifica el concepto del que trata la anotación, y es muy útil para obtener mejoras en las búsquedas, ya que, además, ofrece la posibilidad de extender las búsquedas mediante la inferencia de nuevos conceptos específicos a partir de un concepto más genérico, todo ello dentro del ámbito de las ontologías de referencia que se usen a modo de taxonomías.

- **Basada en conceptos**: Se trata de una técnica donde se combinan las dos técnicas de búsqueda anteriores, utilizando para ello una lista de términos vinculados a un concepto (véase figura 6.14). A partir de los términos de búsqueda introducidos por el usuario, se localizan en la lista de términos y se determina el concepto que les corresponde. Como resultado de la búsqueda se ofrecen los artículos en cuyas anotaciones semánticas figuran dichos conceptos.

- **Consultas en lenguaje natural**. El usuario realiza una consulta expresada en lenguaje natural y el sistema determina los conceptos objeto de búsqueda. La diferencia con la búsqueda basada en conceptos es que aquí no se usan términos de búsqueda aislados ni lista de términos vinculados, sino que se usa directamente el lenguaje natural y el modelo híbrido estudiado en la sección 5.9.4 para determinar estos conceptos. Finalmente se presentan los contenidos web en cuyas anotaciones semánticas figuran los conceptos.

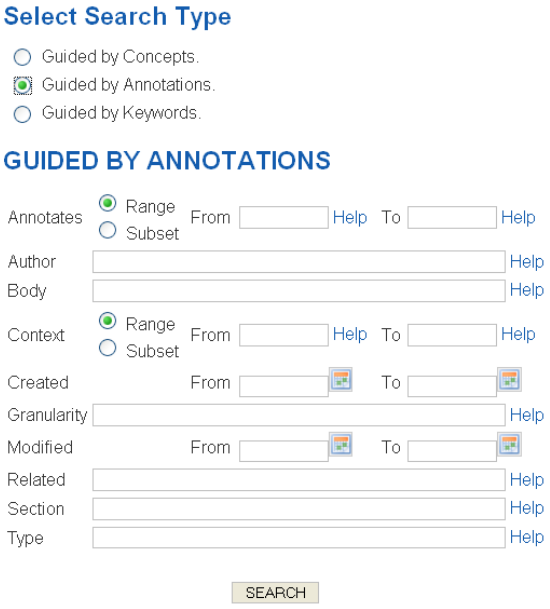


Figura 6.13: Formulario de consulta guiado por anotaciones.

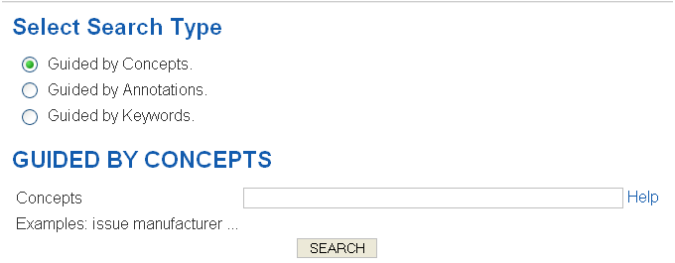


Figura 6.14: Formulario de consulta guiado por conceptos.

6.7. Funciones de Administración o Back-End

En el contexto de los CMS, el término **"Back-End"** se refiere a la parte específica especialmente diseñada para los administradores y los creadores de contenidos, donde se tiene acceso a las herramientas de configuración. El componente *"com_semantic"* dispone de un módulo de administración, ubicado en el menú principal de administración, en la opción **"Components"**, tal y como se muestra en la figura 6.15

Una vez que pulsamos sobre la opción **"Semantic Annotation"** aparecerá una nueva ventana de administración similar a la que se muestra a continuación, en la figura 6.16. La funcionalidad que proporciona es la si-

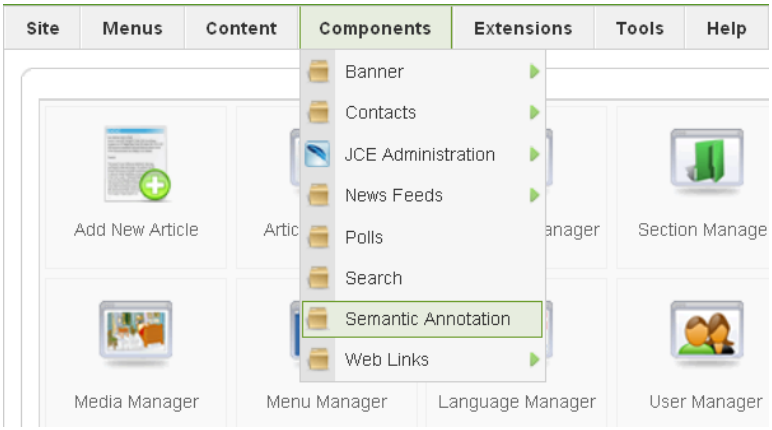


Figura 6.15: Ubicación del módulo de administración.

guiente:

- **Configuración de conceptos**: La pestaña **"Reference Ontologies"** se usa para seleccionar qué ontologías aparecerán en la ventana del selector de ontologías que aparecía en la figura 6.9. Las clases que formen parte de las ontologías seleccionadas podrán ser usados a modo de conceptos cuando se añadan sentencias RDF a una anotación semántica. El aspecto de la pestaña de configuración de conceptos o lo que es lo mismo, de selección de ontologías de referencia, se muestra a continuación en la figura 6.16.

- **Configuración de vocabularios**: La pestaña **"Microformats"** se usa para seleccionar qué propiedades de una ontología pueden participar en el proceso de anotación semántica. Las propiedades que formen parte de las ontologías seleccionadas podrán ser usadas como predicados cuando se añadan sentencias RDF en una anotación semántica. En concreto, los predicados serán accesibles desde la ventana de adición que aparecía en la figura 6.8, en la lista desplegable **"Select property"** que aparece en el campo **"Predicate"**. El aspecto de la pestaña de configuración de vocabularios o, lo que es lo mismo, de selección de microformatos, se muestra a continuación, en la figura 6.17.

- **Precomputación de modelos**: La pestaña **"Precompute Models"** se usa para seleccionar los conceptos que participarán en el proceso de anotación semántica automatizada. Este proceso sólo trabaja con conceptos que pertenezcan a las ontologías seleccionadas aquí, debido a que se necesita un proceso de entrenamiento en el sistema. Se encarga de realizar los cálculos del modelo híbrido estudiados en la sección 5.9.2 sobre los conceptos seleccionados. Estos cálculos serán usados

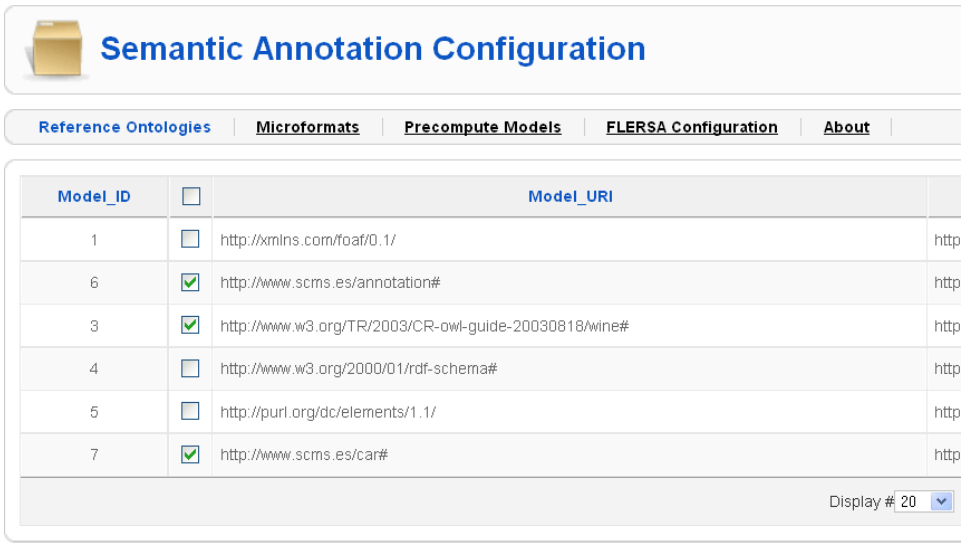


Figura 6.16: Pestaña de selección de ontologías de referencia.

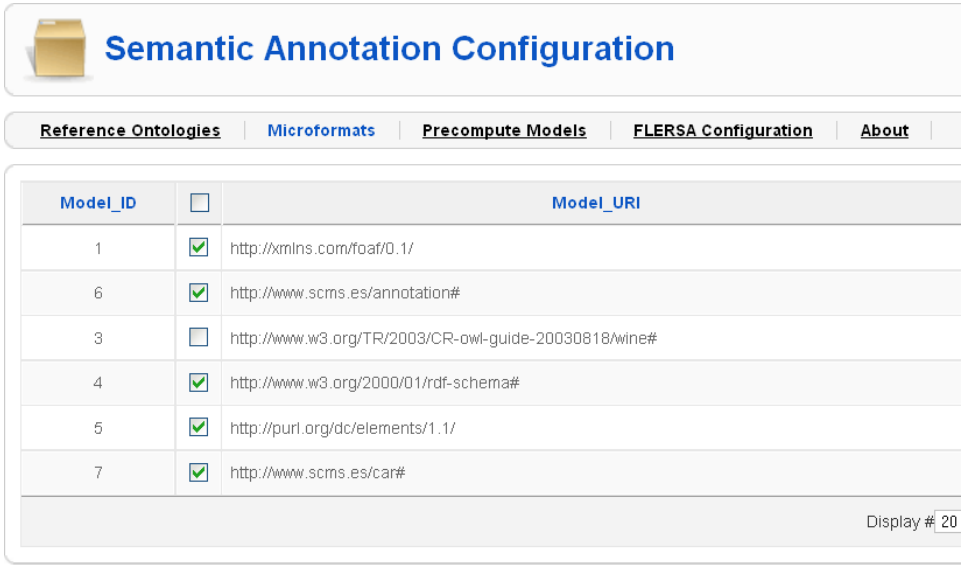


Figura 6.17: Pestaña de selección de microformatos.

cuando se apliquen técnicas de aprendizaje automático presentadas en la sección 5.9.4.

El requisito previo a utilizar esta opción es realizar al menos una anotación manual por concepto que se pretenda usar en el proceso de anotación automatizado. Cada una de las anotaciones debe ser realizada

sobre un fragmento de texto cuya temática verse sobre el concepto que anota, con objeto de vincular las palabras que aparecen en el fragmento de texto con el concepto de la ontología. Esta vinculación es necesaria para realizar el proceso de entrenamiento, ya que éste se encargará de generar los n-gramas de cada concepto y de realizar los cálculos de similitud de forma similar al estudio que se realizó en la sección 5.9.3 del capítulo 5.

El aspecto de la pestaña de configuración de precomputación de modelos se muestra a continuación, en la figura 6.18.

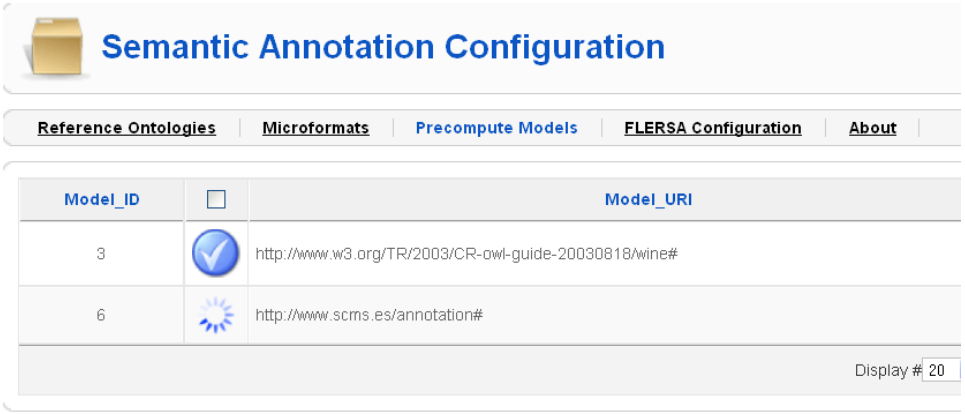


Figura 6.18: Pestaña de selección de precomputación de modelos.

- **Configuración de automatización:** La pestaña **"FLERSA Configuration"** se usa para establecer y examinar las variables del componente, como son el último identificador de anotación y los pesos relativos al proceso de anotación automatizado, tanto local como global. El aspecto de la pestaña de configuración del componente se muestra a continuación, en la figura 6.19.

6.8. Entorno de Pruebas

Una serie de complementos se instalan junto al componente *"com_ semantic"* durante el proceso de instalación con el objetivo de hacerlo operativo desde el primer instante. Se crea una infraestructura de contenidos que permite probar la funcionalidad del componente: Una ontología sobre automóviles, dos conjuntos de artículos para el entrenamiento de sistema de anotación automático y algunos artículos de ejemplo.

La ontología funciona como una taxonomía desde la cual son ordenados

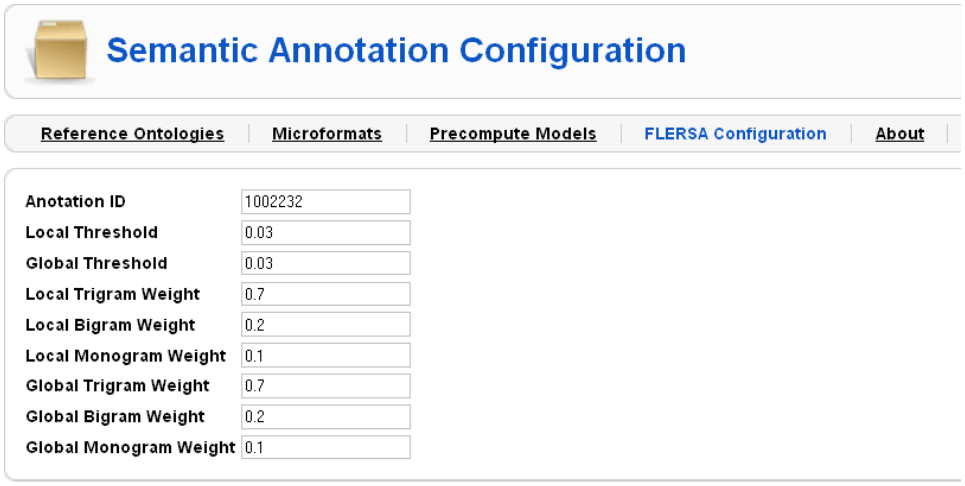


Figura 6.19: Pestaña de configuración de variables del componente.

los conceptos del entorno de pruebas. El concepto **"car_issue"** (aspecto del automóvil) fue creado para proporcionar el marco de trabajo donde operan las anotaciones automatizadas locales. Se han considerado seis características de coches diferentes, esto es: *Styling, interior, engine, driving, costs y safety* (aspecto, interior, motor, conducción, coste y seguridad). Estas características especializadas aparecen en la ontología de automóviles como descendientes del concepto *"car_issue"*. Serán usadas para determinar el aspecto del que trata un párrafo de un documento web durante el proceso de anotación semántica automatizado local.

El concepto **"manufacturer"** fue creado para las anotaciones automatizadas globales. Será usado para determinar la marca de automóvil de la que habla un documento web completo. Se han considerado seis marcas: Audi, BMW, Citroen, Fiat, Ford y Honda. Estas marcas son modeladas también por una ontología de dominio usada como taxonomía de conceptos; éstas aparecen como descendientes de concepto *"manufacturer"*.

Con respecto al conjunto de artículos de entrenamiento, se han creado artículos específicos (formados por unas 20.000 líneas cada uno) para cada uno de los conceptos (categorías) especificados anteriormente, tanto para anotaciones locales como globales. Estos artículos actúan como Corpus, a partir de los cuales son calculados los perfiles de frecuencia de los n-gramas para representar cada concepto.

Finalmente, se ha proporcionado de un conjunto de artículos de prueba para comprobar las funcionalidades del componente.

6.9. Conclusiones

El capítulo versa sobre *"com_ semantic"* el componente para Joomla que se ha desarrollado como prueba de concepto de las soluciones propuestas en la presente tesis. Para el desarrollo del componente se han tenido en cuenta los requisitos estudiados en el capítulo 5, y tiene el doble cometido de, por un lado, ilustrar el funcionamiento de las técnicas de anotación estudiadas, por otro, demostrar cómo es posible aportar nuevas funcionalidades al campo de los CMS para obtener sus equivalentes semánticos, los S-CMS, acortando así la brecha semántica de los CMS hacia su integración en la Web Semántica.

Destacar que el componente está formado por dos grandes partes: Una de administración y otra accesible a los usuarios del CMS. A su vez, en ésta última se pueden encontrar dos herramientas: Una llamada *"FLERSA Annotation Tool"* dedicada a la anotación semántica de contenidos de los artículos del CMS y otra llamada *"Semantic-powered Search"* dedicada a la explotación de la información almacenada en las anotaciones semánticas con objeto de mejorar las búsquedas.

Capítulo 7

Evaluación y Trabajos Relacionados

Basta un poco de espíritu aventurero para estar siempre satisfechos, pues en esta vida, gracias a dios, nada sucede como deseábamos, como suponíamos, ni como teníamos previsto.

Noel Clarasó

Resumen: En esta sección, se evalúa la herramienta de anotación desarrollada y la efectividad de los métodos aplicados. Para ello, se realiza un estudio comparativo de la herramienta de anotación de FLERSA con respecto a otras herramientas de anotación para evaluar dos aspectos: Su funcionalidad y la efectividad del proceso de anotación automático. Para este último aspecto, se lleva a cabo un estudio experimental de los procesos de anotación semántica para cuantificar cómo de bien funcionan los mismos.

7.1. Evaluación

Actualmente, existe un amplio abanico de herramientas para la producción de etiquetas semánticas. Algunas de ellas proporcionan marcado de páginas web pero no soportan la generación automática de anotaciones semánticas. Tampoco soportan la evolución de los documentos anotados con ellas, debido a que suelen estar basados en la tecnología XPointer. Este tipo de herramientas no están bien integradas en los entornos de publicación

dotados de semántica, por lo que la información semántica no se explota adecuadamente. Otras herramientas soportan la generación automatizada de anotaciones semánticas, pero este tipo de herramientas ofrecen generalmente un enfoque centrado en la plataforma en lugar de en el usuario. También disponen de complejos procesos de instalación y algunas están convirtiéndose en obsoletas. A continuación se realiza un estudio más detallado de toda esta problemática y de algunas de las herramientas de anotación semántica más representativas.

Partiendo de la base de los requisitos señalados en el trabajo de Uren et al. (2006) y tomando en cuenta los aspectos que se han expuesto anteriormente, se ha realizado un estudio comparativo de cara a evaluar la herramienta de anotación de FLERSA.

Las herramientas que se comparan son aquellas consideradas más representativas, junto con aquellas que se aludieron en el artículo de Uren et al. (2006) y que han persistido en el tiempo, todas ellas estudiadas en la sección 4.4 del capítulo 4. En orden alfabético, son: Amaya, Dbin, FLERSA, GoNtogle, KIM, MnM, Ontomat, Smore, Soboleo y Text2Onto.

Se han tenido en cuenta 15 características en el proceso de comparación, varias de las cuales fueron especificadas como requisitos que deben cumplir los S-CMS; el resto de características están relacionadas con las características que ofrecen las herramientas de anotación. Son estas:

1. **Formato de anotación:** Indica los lenguajes utilizados para almacenar los metadatos de las anotaciones semánticas. En el requisito 1 de la sección 5.3 se establece el uso de estándares abiertos recomendados por el W3C tales como: XML, RDF, RDFa y OWL.

2. **Entorno colaborativo:** En el requisito 2 de la sección 5.3 se establece que el sistema de anotación debe facilitar la colaboración entre usuarios y expertos con el objetivo de reutilizar el conocimiento adquirido.

3. **Diseño centrado en el usuario:** El entorno en el que los usuarios anotan los documentos debe estar integrado con el entorno en el que trabajan. Esta característica también se recoge en el requisito 2 de la sección 5.3.

4. **Formato de los documentos:** Indica los diferentes formatos con los que las herramientas de anotación son capaces de trabajar. En el requisito 4 de la sección 5.3 establece que los sistemas de anotación deben contar con soporte para trabajar con formatos de documento heterogéneos.

5. **Consistencia de anotaciones:** Esta característica está relacionada con la consistencia de las anotaciones en los documentos cuando la

información contenida en ellos es revisada, borrada o movida, tal y como se establece en el requisito 5 de la sección 5.3. Las herramientas de anotación basadas en la tecnología XPointer para delimitar el texto que se está anotando presentan problemas de consistencia.

6. **Almacenamiento:** Indica dónde se almacenan las anotaciones. Almacenar las anotaciones semánticas embebidas dentro del documento donde se hacen, ayuda a mantener la consistencia frente a cambios y nuevas versiones del documento, tal y como queda establecido en el requisito 6 de la sección 5.3.

7. **Perfil de usuario:** Indica la facilidad de uso de la herramienta. Los tipos considerados son: Experto, avanzado y estándar. Este aspecto está fuertemente interrelacionado con la característica número 11, la cuál se estudiará más adelante.

8. **Entorno:** Explica algunos de los aspectos técnicos relativos a la herramienta de anotación semántica considerada, tales como tipo de aplicación y lenguaje de programación usado para desarrollar la herramienta.

9. **Fecha de la última versión:** Se refiere a la fecha de la última versión de la herramienta. Se usa para identificar cómo de reciente es la herramienta.

10. **Tipos de anotación:** Indica los tipos de anotación soportados. Los cuatro tipos que existen fueron comentados en la sección 4.2, y son:

    - Global, cuando la anotación se refiere al documento completo.
    - Local, cuando la anotación se refiere a un fragmento de texto.
    - Manual, cuando las anotaciones son hechas por los usuarios.
    - Automática, cuando las anotaciones son hechas por el sistema siguiendo alguna técnica de aprendizaje.

11. **Estructura de las anotaciones:** Indica la complejidad estructural de las anotaciones. Mientras que algunas herramientas proporcionan etiquetas para realizar anotaciones, otras utilizan ontologías. Basándonos en esto, la estructura de etiquetas puede ser hecha por usuarios estándar y fácilmente mapeada en RDF. Las ontologías complejas no están aconsejadas cuando los usuarios no son expertos en tecnologías de la Web Semántica debido a la dificultad del proceso.

12. **Vocabulario de anotación:** Este aspecto se refiere a la expresividad de la herramienta. Cuando se está anotando, las herramientas deben ofrecer al usuario un vocabulario controlado y una taxonomía tipo. Es recomendable dirigir a los usuarios hacia el uso de vocabularios controlados para salvar el problema de la brecha semántica.

13. **Papel de las ontologías:** Describe cómo son usadas la ontologías: De una forma estructural o como ayuda al proceso de anotación semántica. Las ontologías estructurales son usadas para describir los recursos anotados. Las ontologías de apoyo son usadas a modo de base de conocimiento para ayudar al proceso de anotación semántica; los usuarios proporcionan elementos de anotación y después los mapean con respecto a ontologías de referencia. En el requisito 3 de la sección 5.3 establece que el sistema de anotación sea capaz de trabajar con múltiples ontologías.

14. **Análisis de la automatización:** Especifica las bases teóricas para los procesos de anotación semántica cuando éstos funcionan en modo automático. En el requisito 7 de la sección 5.3 se establece la obligatoriedad de proporcionar herramientas de marcado automatizado.

15. **Aprendizaje:** Indica cuándo la herramienta tiene algún tipo de procedimiento de aprendizaje para mejorar el sistema automatizado de anotación a lo largo del tiempo.

Que sepamos, no existe todavía ninguna herramienta que reúna todas las características expuestas anteriormente. La principal motivación para desarrollar la herramienta FLERSA fue intentar tener en cuenta tantos aspectos como fuera posible de los estudiados con anterioridad.

En las páginas siguientes, las tablas 7.1 y 7.2 proporcionan una comparativa entre las herramientas de anotación semántica consideradas y en ellas se han resumido sus características principales.

Tabla 7.1: Comparativa de herramientas I.

TOOL	**1 Formato de anotación**	**2 Entorno Colaborativo**	**3 Diseño centrado en el usuario**	**4 Formatos de documentos**	**5 Consistencia**	**6 Almacenamiento**	**7 Perfil de usuario**	**8 Entorno**	**9 Ultima versión**
Amaya (cliente Annotea)	RDF	Si	Navegador y editor	HTML, XHTML and XML	No	Separado; Servidor	Estandard	Standalone, C++	2009
DBin	RDF	Si, P2P	Navegador	URIs	No	Local	Avanzado	Standalone, Java	2008
FLERSA	RDFa, RDF	Si, CMS	Integratedo en el CMS	HTML e imágenes	Si	Incrustado; Servidor	Estándar	Componente de Joomla, PHP	2011
GoNTogle	OWL, RDF(S)	Si	Interfaz gráfica	PDF, ODT, DOC, SXW, RTF, TXT	No	Servidor	Estándar	Standalone, Plugin de Openoffice	2010
KIM	RDF(S), OWL	Si	Varios plug-in front ends	HTML	No	Servidor	Experto	Standalone, Java	2010
MnM	RDF	No	Navegador	HTML, TEXT	No	Incrustado	Estándar	Standalone, Java	2004
Ontomat	RDF, OWL	Si	Drag & drop, crear & anotar	HTML y multimedia	No	Servidor, Incrustado, fichero	Avanzado	Standalone	2006
Smore	RDF(S)	Si	Navegador y editor	HTML e imagesne	No	Servidor	Avanzado	Standalone	2005
SOBOLEO	RDF	Si	Navegador	URLs	No	Servidor	Estándar	Web, JSP	2007
Text2Onto	OWL	Si	Inferza gráfica	TXT, PDF, HTML	No	Servidor Gate	Experto	Plugin, Java	2010

Tabla 7.2: Comparativa de herramientas II.

TOOL	**10 Tipo de anotación**	**11 Estructura de anotación**	**12 Vocabulario**	**13 Ontologías**	**14 Análisis en automatización**	**15 Aprendizaje**
Amaya (Annotea client)	Manual; Local	Esquema Annotea	Fijo	Estructural	No	No
DBin	Manual	Etiquetas	Configurable	Estructural; Apoyo	No	No
FLERSA	Global; Local; Manual; Automático	FLERSA ontology: Variante de Annotea	Configurable FLERSA ontology	Estructural; Apoyo	VSM; n-gramas	Supervisado
GoNTogle	Global; Local; Manual; Automático	Basado en ontologías	Configurable	Estructural; Apoyo	KNN clasificación	Supervisado
KIM	Manual; Automático	Ontologías PROTON y WKM	Configurable	Estructural	Coincidencia; etiquetado POS; reconocimiento de entidades	No
MnM	Manual; Automático	Basado en ontologías	Configurable	Estructural	Etiquetado POS; reconocimiento de entidades	Supervisado
Ontomat	Manual; Automático	Ontobroker	Configurable	Apoyo	PANKOW; Amilcare	Supervisado
Smore	Manual; Automático	Basado en ontologías	Configurable	Apoyo	Screen Scrapper	No
SOBOLEO	Manual; Global	Etiquetas	Configurable SKOS Core	Apoyo	No	No
Text2Onto	Automático	Probabilistico	Configurable Wordnet	Estructural	Linguístico - GATE	No supervisado

Las tablas revelan inmediatamente que FLERSA tiene características únicas, tales como 1) incrusta las anotaciones dentro del documento en formato RDFa, y 5) consistencia de las anotaciones frente a los cambios y 6) almacenamiento dual de anotaciones.

Haciendo un estudio comparativo más detallado, cabe destacar las siguientes características que distinguen a FLERSA de las demás herramientas:

- Evita el problema conocido como "La Web profunda" ("Deep Web" en inglés) para las anotaciones en los documentos. Los indexadores de los motores de búsqueda pueden acceder a la información semántica almacenada en los documentos anotados con la herramienta, ya que las anotaciones están incrustadas dentro de los documentos en formato RDFa (característica 1).

- La herramienta usa el lenguaje de marcas RDFa para las anotaciones incrustadas dentro del documento y el lenguaje RDF se usa para almacenar la misma información semántica en el lado servidor (almacenamiento dual, características 1 y 6). Los formatos RDF(S) y OWL son usados por otras herramientas como formatos de anotación en el servidor; estos formatos tienen mayor expresividad que RDF y se suelen usar para razonamiento automático de información, aunque para propósitos de anotación es suficiente con usar RDF. La herramienta FLERSA también permite que las anotaciones en RDF del lado servidor, sean combinadas con ontologías expresadas en formato OWL para inferir nuevo conocimiento (característica 12).

- Ofrece una herramienta colaborativa, provista por el CMS subyacente (característica 2).

- Proporciona un entorno CMS centrado en el usuario (característica 3).

- Soporta formato nativo HTML para páginas web (característica 4).

- Ofrece consistencia de las anotaciones frente a cambios, proporcionada por el uso de la técnica de Definición de Rangos Flexibles (en lugar de XPointer) durante el proceso de marcado del documento (característica 5).

- Proporciona almacenamiento dual de anotaciones definidas en un documento, tanto en el lado servidor como incrustadas en el mismo documento donde se realiza el marcado (característica 6).

- Facilidad de uso a nivel de perfil de usuario estándar, o avanzado a lo sumo (característica 7).

- Se trata de un entorno de trabajo ligero para la infraestructura CMS. Aunque FLERSA se ha implementado sobre Joomla como prueba de concepto, puede ser fácilmente adaptado para que trabaje con cualquiera de los CMS más populares tales como Drupal (Mercer, 2008), Typo3 (Peacock, 2007) y Wordpress (Hayder y Silver, 2009) (característica 8).

- Proporciona una herramienta reciente y actualizada (característica 9).

- Permite los siguientes tipos de anotación: Global, local, manual y automática (característica 10).

- Permite anotaciones basadas en ontologías; la estructura de anotación está basada en el esquema de anotación Annotea y se denomina ontología FLERSA (característica 11).

- Ofrece vocabularios controlados configurables mediante ontologías (característica 12).

- Herramienta basada en ontologías a nivel estructural y de apoyo. La herramienta establece enlaces entre los conceptos de las ontologías de apoyo y los fragmentos de texto que se refieren a esos conceptos (característica 13).

- Proporciona generación automática de anotaciones semánticas basadas en un enfoque híbrido que usa el Modelo de Espacio Vectorial y n-gramas como base teórica. Estas técnicas de aprendizaje son usadas para anotar automáticamente la información específica de dominio en grandes repositorios de información (característica 14).

- Permite aprendizaje supervisado, a través de un enfoque "paga según recibas", donde el sistema comienza con un Corpus elemental. Conforme crece el contenido web y las anotaciones manuales/automáticas son validadas por el ingeniero de conocimiento, el Corpus mejora con el tiempo y se obtiene un incremento en la efectividad global del sistema de anotación.

- Se mezcla la recuperación de información tradicional con la semántica basada en ontologías (característica 15).

Se puede considerar la herramienta de anotación de FLERSA como una herramienta muy valiosa debido a que la combinación de todas sus características, expuestas anteriormente, diferencia a esta herramienta del resto.

7.2. Evaluación de los Procesos de Anotación Automáticos

En esta sección se evalúa la calidad y efectividad del proceso automático de anotación semántica. Se ha construido un entorno de evaluación para este propósito, compuesto de una ontología de dominio, un conjunto formado por cincuenta artículos de prueba y dos Corpus.

Como se ha explicado anteriormente, la ontología funciona como taxonomía de conceptos y se usa para establecer enlaces entre conceptos y anotaciones semánticas conforme al asunto del que traten. Su estructura se compone de dos clases o conceptos principales: **"Car_issue"** y **"manufacturer"**. Por un lado, la clase *"car_issue"* tiene, a su vez, seis subclases para modelar las características de diferentes coches, y son: *Estilo, interior, motor, conducción, coste y seguridad.* Por otro lado, la clase *"manufacturer"* dispone de otras seis subclases con objeto de modelar marcas de coches, de las cuales se han considerado para la evaluación las siguientes: *Audi, BMW, Citroen, Fiat, Ford y Honda.*

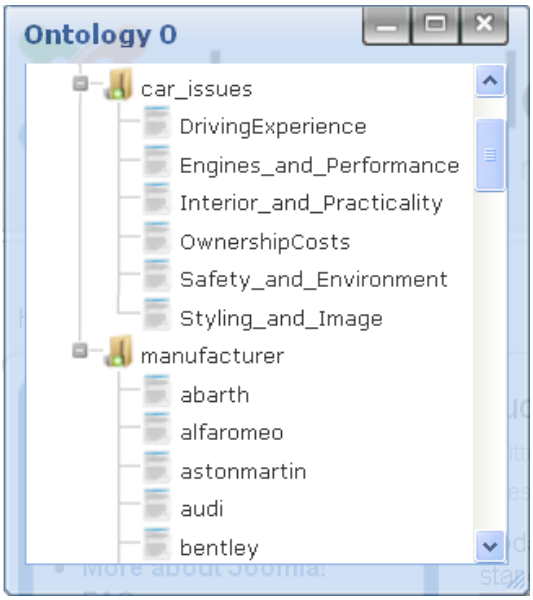


Figura 7.1: Ontología de dominio.

La figura 7.1 muestra el aspecto jerárquico de la ontología de dominio que se ha usado durante el proceso de evaluación.

El conjunto de artículos de prueba se compuso de 50 artículos que trataban sobre conceptos de coches tales como marcas, modelos y características.

También se incluyeron artículos sobre conceptos externos a los considerados para la evaluación; fueron considerados conceptos sobre tipos de vino y de pizza con objeto de comprobar cómo se comportaba el sistema con información de otros ámbitos. La información de los artículos fue tomada de al URL siguiente: `http://www.autoexpress.co.uk`

Con respecto a los dos Corpus, se usó uno global para la anotación automática a nivel de artículo (texto completo del documento web). Éste se compone de información seleccionada, recopilada por el doctorando en la tesis actuando en el rol de ingeniero de conocimiento, que se usará para determinar la marca del coche del que se habla en un artículo. El Corpus local también ha sido elaborado por el doctorando y se usa para la anotación automática del documento a nivel de párrafo; se usa para determinar de qué característica sobre coches, de entre las citadas anteriormente, habla cada párrafo.

En cuanto al proceso de evaluación, comenzó con los Corpus, es decir, un conjunto preexistente de textos a partir de los cuales modelar los distintos conceptos del entorno de evaluación: Marcas y características de coches. Para cada uno de los conceptos de la ontología de dominio se dimensionaron páginas web de unos 20K de tamaño cada una. A partir de ellas, fueron precomputados los perfiles de frecuencia de sus n-gramas para representar cada uno de los conceptos (categorías). Después se realizaron anotaciones automáticas a nivel local y global sobre los 50 artículos de prueba y se anotaron los resultados obtenidos. Conforme el proceso de anotación automática progresaba y, siguiendo la filosofía "paga según recibas", las anotaciones semánticas fueron validadas por el ingeniero de conocimiento; así el Corpus fue actualizado cada 5 artículos y las anotaciones semánticas correctas fueron incorporadas al Corpus.

Las pruebas realizadas que se muestran a continuación en la tabla 7.3 fueron llevadas a cabo a partir de 50 artículos de prueba y 293 anotaciones semánticas automáticas diferentes.

En cuanto al diagnóstico de la evaluación cabe decir que desde el principio el sistema automático de anotación semántica funcionó bastante bien aunque tuvieron lugar algunos errores. Conforme fue actualizándose el Corpus con las anotaciones validadas por la figura de ingeniero de conocimiento, fue creciendo la efectividad global del sistema de anotación semántica.

Los resultados obtenidos con la prueba de concepto son satisfactorios, como así lo justifica el ratio de éxito obtenido al aplicar el enfoque híbrido en el sistema automático de anotación semántica.

Tabla 7.3: Tabla de resultados de prueba para la anotación automática.

Concepto	**Tipo**	**Caracteres**	**Monogramas**	**Bigramas**	**Trigramas**	**Correcto**	**Falso -**	**Falso +**
Estilo	Local	22783	1203	3148	3818	84.62 %	15.38 %	0.00 %
Interior	Local	26100	1169	3340	4234	93.10 %	6.90 %	0.00 %
Motor	Local	25680	1115	3205	4198	84.21 %	12.28 %	3.51 %
Conducción	Local	23486	1094	3031	3770	87.50 %	10.00 %	2.50 %
Coste	Local	21968	1080	3011	3727	88.46 %	0.00 %	11.54 %
Seguridad	Local	19880	997	2629	3219	82.35 %	11.76 %	5.88 %
Audi	Global	194	21	39	40	94.74 %	5.26 %	0.00 %
BMW	Global	341	44	79	86	100.00 %	0.00 %	0.00 %
Citroen	Global	585	43	85	92	100.00 %	0.00 %	0.00 %
Fiat	Global	852	64	129	154	100.00 %	0.00 %	0.00 %
Ford	Global	644	60	107	111	85.71 %	0.00 %	14.29 %
Honda	Global	336	36	59	61	100.00 %	0.00 %	0.00 %
Promedio		11904	577	1571	1959	**91.72 %**	**5.13 %**	**3.14 %**

La tabla 7.3 cuenta con datos dispuestos en nueve columnas:

- La primera columna describe el concepto que se está considerando
- La segunda describe a qué nivel es utilizado el concepto por el sistema automatizado dentro de los artículos de prueba, *"local"* significa que se trata de un concepto que el proceso automatizado asigna a nivel de párrafo, indicando que el párrafo trata sobre ese concepto, y *"global"* que el concepto se asigna a nivel global de artículo, indicando que el concepto es el tema principal de un artículo considerado en su globalidad.
- Las columnas 3, 4, 5 y 6 muestran información acerca del volumen de información que se está manejando, a distintos niveles, en los Corpus de entrenamiento de los distintos conceptos
- Finalmente, en las columnas 7, 8 y 9 se muestran resultados obtenidos, siendo la columna 7 la que muestra los resultados de ejecución correcta de anotaciones semánticas automatizadas, mientras que las columnas 8 y 9 muestras los resultados donde se apreciaron errores de ejecución.

Cabe puntualizar, en cuanto al significado de las estadísticas que se ofrecen en la tabla, que se ha considerado *"correcto"* el resultado de las anotaciones semánticas automatizadas realizadas sobre textos cuyo contenido trata sobre el concepto que se anota, es decir, el sistema de anotación ha relacionado correctamente el texto con el concepto implícito que se debate en el mismo. Se ha considerado el resultado *"falso negativo"* cuando el proceso de anotación semántica automatizada no ha vinculado un texto con ningún concepto, cuando debería de haber sido así. Se ha considerado el resultado *"falso positivo"* cuando el proceso de anotación semántica automatizada ha vinculado un texto con un concepto y el ingeniero de conocimiento dictamina que el concepto que le correspondería debería ser otro.

7.3. Conclusiones

En este capítulo se ha realizado una evaluación de la herramienta de anotación semántica, la cual constituye la piedra angular de la prueba de concepto de S-CMS desarrollado en FLERSA.

La evaluación se ha realizado a dos niveles: A modo de estudio comparativo de la funcionalidad que aporta FLERSA con respecto a otras herramientas de anotación semántica, y un segundo estudio sobre la efectividad del proceso de anotación semántica automatizada sobre un entorno de pruebas.

Los resultados de las dos evaluaciones han sido satisfactorios. Por un lado, ha quedado constancia de que sólo en la herramienta de anotación que proporciona FLERSA se reúnen todos los requisitos y características deseables expuestas en este capítulo. Por otro lado, las pruebas realizadas en el entorno de evaluación han confirmado, mediante la obtención de altos porcentajes en la correcta anotación de artículos, la efectividad de los métodos utilizados para la generación automatizada de anotaciones semánticas.

Capítulo 8

Conclusiones y Trabajo Futuro

Cuando llegamos a la meta, creemos que
el camino ha sido el bueno.

Paul Valéry

Resumen: En este último capítulo se realiza un resumen de todas las aportaciones que se han ido realizando a lo largo de la tesis. También se explica la situación en que se encuentra la línea de investigación emprendida y las perspectivas de trabajo futuro que se presenta en esta dirección.

8.1. Introducción

Llegados a este punto, resulta muy satisfactorio recordar su punto de inicio y ser consciente de que consideramos que se ha conseguido el objetivo. El objetivo inicial parecía simple: Hacer una aportación enriqueciendo los CMS con las tecnologías de la Web Semántica. El objetivo resultó ser más complejo, en particular, concretar las propuestas en la prueba de concepto que, por otra parte, ha ayudado a refinarlas. En este capítulo se condensan las aportaciones realizadas en el contexto de la línea de investigación.

8.2. Conclusiones

En esta memoria de Tesis, se ha realizado la propuesta de cómo convertir un sistema web de gestión de contenidos en su equivalente semántico para así

obtener los beneficios que predica la Web Semántica. Como prueba de concepto se ha desarrollado un componente de Joomla llamado *"com_ semantic"*, el cual también ha sido presentado en la presente tesis.

En la herramienta de anotación del sistema FLERSA se han desarrollado todas las características deseables para un sistema de anotación estudiadas en la sección 5.3, entre las que destacan: Sistema centrado en el usuario, entorno de trabajo ligero, soporte para anotaciones manuales y automáticas, sigue el enfoque "paga según recibas", evita el problema de "la Web profunda" en sus anotaciones, ofrece anotaciones basadas en ontologías y proporciona funciones para recuperación de información basada en la semántica.

También se ha propuesto un proceso de anotación semántica, tanto manual como automatizado. En particular, cabe destacar las siguientes características importantes del sistema:

- La técnica de **Definición de Rangos Flexibles** para el marcado de texto en documentos web: Su principal ventaja es la de permitir la evolución de los documentos web que disponen de anotaciones semánticas definidas con esta técnica. También posibilita el almacenamiento dual de las anotaciones semánticas definidas en el documento, tanto en el lado servidor como dentro del documento que se marca.
- La **ontología FLERSA-Ontology** que da soporte a la definición de anotaciones de infraestructura. Se ha ilustrado cómo usar una ontología base, basada en la estructura Annotea, para la creación de anotaciones semánticas, justificándose los beneficios que proporciona su uso. También se considera la faceta de cómo gestionar ontologías de forma que permitan el uso de taxonomías de conceptos para extender los vocabularios de anotación, e incluso para la creación de instancias o individuos de conceptos que aparecen en la Base de Conocimiento.
- El **enfoque híbrido modelo de espacio vectorial + ngramas** para determinar los conceptos de los cuales tratan las anotaciones semánticas automatizadas.
- El componente **com_semantic**, una distribución de código libre en la que se implementan las herramientas de anotación y de búsqueda avanzada con apoyo de ontologías presentadas en la presente tesis.

Llegado este punto, podemos cuestionarnos cómo contribuye la herramienta FLERSA a realizar una aproximación de la Web actual en la dirección que marca la Web Semántica. La respuesta es que en caso de que FLERSA recibiera una amplia difusión por todos los sitios web donde se usan CMS, gracias a su principal característica, la de almacenar en RDFa las anotaciones semánticas incrustadas en los documentos que se anotan, los indexadores

de los motores de búsqueda Web tendrían a su disposición los metadatos que generan los S-CMS, a los cuales se les podría sacar beneficio para mejorar las búsquedas. Los buscadores web deberían incorporar una infraestructura para trabajar con ontologías, así como ontologías de dominio que funcionaran a modo de "piedra roseta" permitiendo el mapeo de distintos conceptos locales en una taxonomía de conceptos estandarizada.

8.3. Trabajo Futuro

En cuanto al trabajo futuro, nos planteamos diversos frentes:

- En primer lugar se pretende refinar, en la medida de lo posible, el proceso de anotación automático para que su efectividad sea próxima al 100 %. También se plantea la posibilidad de definir nuevos procesos de anotación, como es el caso de los procesos semi-automáticos y comprobar qué ventajas ofrecen frente a los procesos manuales y/o automatizados.
- Por otro lado, sería interesante realizar una adaptación del código fuente del componente desarrollado a la nueva versión de Joomla, la 1.7, con objeto de conseguir que tenga mayor difusión en la comunidad que utiliza Joomla.
- También sería posible migrar la implementación de *"com_semantic"*, el componente para Joomla, a otros CMS como Drupal[1], Typo3[2] o Wordpress[3] y permitir así la transformación de los CMS en sus equivalentes semánticos, obteniéndose los beneficios que la Web Semántica les puede aportar.
- Actualmente se sigue investigando sobre cómo realizar una mejor explotación de las anotaciones semánticas. Se está trabajando en el diseño de un método híbrido y unificado de búsqueda que proporcione una combinación flexible de los métodos de búsqueda tradicionales, basados en palabras clave, con los métodos de búsqueda semánticos, basados en anotaciones semánticas y metadatos.
- Por último, se pretenden realizar estudios experimentales sobre el funcionamiento de la herramienta sobre entornos específicos, como por ejemplo en fisiología, educación y sanidad.

[1] `http://drupal.org`
[2] `http://typo3.com`
[3] `http://www.wordpress.org`

Apéndice A

Publicaciones Derivadas de la Tesis

En este anexo se listan las publicaciones generadas como resultado del desarrollo de la tesis. Están ordenadas por fecha, en forma descendente. Para cada publicación se presenta un breve resumen, se indica los capítulos de la tesis donde ha quedado reflejado su contenido, también se muestran los indicios de calidad del foro de publicación.

Como resumen, se han producido: Una publicación indexada CORE B, dos publicaciones CORE C, una en conferencia nacional de referencia; asimismo otra publicación está en proceso de revisión. Se puede considerar que todo el contenido de la tesis está publicado o enviado a publicar. Todas las publicaciones se han realizado en conferencias o revistas del área de la tesis o bien con secciones dedicadas a ella y han sido revisadas por, al menos, tres revisores cada una. Este es el medio por el cuál se le está dando difusión al contenido de la tesis, tanto a nivel nacional como internacional.

Las publicaciones son las siguientes:

- Navarro-Galindo, J. L., Samos, J. y Alférez Muñóz, M. J. A Hybrid Approach to Text Categorization Applied to Semantic Annotation.
    - Artículo aceptado con fecha 17/05/2012 en el congreso DEXA 2012[1], indexado en CORE nivel B[2] y especializado en Bases de Datos y Aplicaciones de Sistemas Expertos.
    - El artículo presenta un enfoque híbrido (Modelo de Espacio Vectorial combinado con n-gramas) como una nueva técnica para la

[1] http://www.dexa.org/accepted_papers

[2] Siguiendo la clasificación de revistas de la asociación CORE (Computing Research and Education, http://core.edu.au)

categorización de texto que puede ser aplicada en la anotación semántica automatizada. También realiza una evaluación de la técnica dentro del ámbito de la fisiología como dominio específico de conocimiento.

- El artículo tiene correspondencia con la sección 9 del apartado 5 de la presente tesis, lugar donde se estudia del proceso automático de anotación semántica en la herramienta FLERSA.

- Navarro-Galindo, J. L. y Samos, J. The FLERSA Tool: Adding Semantics to a Web Content Management System. International Journal of Web Information Systems (IJWIS), vol. 8(1), páginas 73-126, 2012. ISSN 1744-0084.

  - Revista indexada CORE nivel C[3] especializada en Sistemas de Información Web.
  - El artículo proporciona un punto de partida para futuras investigaciones en las que los principios y técnicas de la herramienta FLERSA se puedan aplicar a cualquier WCMS. El desarrollo de la herramienta muestra que es posible construir un WCMS semántico mediante la combinación de componentes semánticos y otros recursos tales como ontologías y tecnologías emergentes, en las que se pueden incluir XML, RDF, RDFa y OWL.
  - En este amplio artículo de 53 páginas se describen todos los aspectos esenciales de la tesis. Tiene correspondencia con los apartados 4, 5, 6 y 7 de la misma, que se ocupan de la anotación semántica de documentos web, del estudio de la herramienta FLERSA y de la evaluación de la misma.

- Navarro-Galindo, J. L. y Samos, J. FLERSA: Soporte a la Definición de Anotaciones y Búsquedas Semánticas en un CMS. En Actas de las XVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2011), páginas 101-114. A Coruña, Spain, 2011. ISBN 978-84-9749-486-1. http://www.sistedes.es/jornadas2011/jisbd.htm.

  - Foro de referencia y encuentro para investigadores y profesionales de España, Portugal e Iberoamérica en los campos de la Ingeniería del Software y de las Bases de Datos.
  - En el artículo se presenta la herramienta de anotación semántica FLERSA y se muestran las posibilidades de búsquedas que los metadatos ofrecen: guiada por las anotaciones, guiadas por conceptos y basada en lenguaje natural.

[3]Siguiendo la clasificación de revistas de la asociación CORE (Computing Research and Education, http://core.edu.au)

- En general, tiene correspondencia directa con el apartado 5 de la tesis, y en particular, con la sección 10 encargada de cómo realiza FLERSA la recuperación de información. También tiene correspondencia con el apartado 4 ya que en el artículo se presentan de forma introductoria conceptos relacionados con la anotación semántica de documentos web.

- Navarro-Galindo, J. L. y Samos, J. Manual and Automatic Semantic Annotation of Web Documents: The FLERSA Tool. En 12th International Conference on Information Integration and Web-based Applications and Services (iiWAS 2010), vol. 1, páginas 540-547. ACM, Paris, France, 2010. ISBN 978-1-4503-0421-4.

  - Conferencia indexada en CORE nivel C[4] especializada en Integración de Información y en Servicios y Aplicaciones Web.
  - En el artículo se presenta FLERSA como una herramienta desarrollada sobre un CMS donde es posible la anotación de documentos web de forma manual y automatizada. En la anotación manual se presenta la técnica de marcado flexible de rangos de texto, la cual permite la evolución de los documentos web anotados más efectivamente que otras técnicas como XPointer. En la anotación automatizada se presenta un enfoque híbrido (modelo de espacio vectorial combinado con n-gramas) con el cual determinar los conceptos de los que trata el contenido de un documento web.
  - Tiene correspondencia directa con el apartado 5 de la tesis encargado del estudio de los aspectos esenciales para la definición de un CMS semántico. También tiene correspondendia con el apartado 4 ya que en el artículo se presentan de forma introductoria conceptos relacionados con la anotación semántica de documentos web.

- Navarro-Galindo, J. L. y Samos, J. Flexible Range Semantic Annotations based on RDFa. En 27th British National Conference on Databases: Data security and security data (BNCOD 2010), páginas 122-126. Springer-Verlag, Dundee, United Kingdom, 2010. ISBN 978-3-642-25703-2.

  - Conferencia indexada en CORE nivel B[3] especializada en formatos de datos y en Sistemas de Información en general.
  - En el artículo se presenta nuestra técnica de marcado flexible de rangos de texto basada en el estándar RDFa, la cual permite la evolución de los documentos anotados con ella de forma más eficiente que si se usase la tecnología XPointer.

[4]Siguiendo la clasificación de revistas de la asociación CORE (Computing Research and Education, http://core.edu.au)

  - Tiene correspondencia directa con los apartado 4 y 5 de la tesis. En particular, se corresponden con las secciones 7 y 8 del apartado 5 encargadas de presentar los problemas que se presentan a la hora de anotar un documento web y como nuestra técnica de anotación mediante rangos flexibles de texto los resuelve.

- Navarro-Galindo, J. L. y Samos, J. Una panorámica actual de software para trabajar con ontologías. En I Simposio en Desarrollo Software (SDS07), páginas 101-116. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada, Granada, Spain, 2007. ISBN 978-84-96856-17-2.

  - Simposio en Desarrollo de Software desarrollado por la Universidad de Granada para los doctorandos que optan al título de master y/o DEA.
  - En el artículo se presenta una panorámica de las herramientas software más representativas que existían en 2007 para trabajar con la Web Semántica.
  - Tiene correspondencia con la sección 4 del apartado 3 de la tesis encargada de introducirnos en la Web Semántica y aspectos tales como sus principios, herramientas y situación.

Bibliografía

Y así, del mucho leer y del poco dormir,
se le secó el celebro de manera que vino
a perder el juicio.

Miguel de Cervantes Saavedra

Adida, B. y Birbeck, M. RDFa primer 1.0 embedding RDF in XHTML. W3c recommendation, W3C, 2007. http://www.w3.org/TR/2007/WD-xhtml-rdfa-primer-20071026/.

Auer, S. pOWL - A Web Based Platform for Collaborative Semantic Web Development. En *Proceeding of 1st Workshop Scripting for the Semantic Web (SFSW'05), Hersonissos, Greece, May 30*. CEUR Workshop Proceedings, 2005.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D. y Patel-Schneider, P. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003. ISBN 0521781760.

Beckett, D. RDF/XML Syntax Specification (Revised). W3c recommendation, W3C, 2004a. Http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/.

Beckett, D. RDF/XML Syntax Specification (Revised). Informe técnico, 2004b. http://www.w3.org/TR/rdf-syntax-grammar/.

Bergin, J. Building Graphical User Interfaces with the MVC Pattern. 2007. http://pclc.pace.edu/~bergin/mvc/mvcgui.html.

Berglund, A., Boag, S., Chamberlin, D. D., Fernández, M. F., Kay, M., Robie, J. y Siméon, J. XML Path Language (XPath) 2.0. World Wide Web Consortium, Recommendation REC-xpath20-20070123, 2007. http://www.w3.org/TR/2007/REC-xpath20-20070123.

Bergman, M. K. White Paper: The Deep Web. Surfacing Hidden Value. *The Journal of Electronic Publishing*, vol. 7(1), página online, 2001. http://dx.doi.org/10.3998/3336451.0007.104.

BERNERS-LEE, T., FIELDING, R. y MASINTER, L. RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. 2005. http://rfc.net/rfc3986.html.

BERNERS-LEE, T., HENDLER, J. y LASSILA, O. The Semantic Web. *Scientific American*, vol. 284(5), páginas 34–43, 2001. http://www.scientificamerican.com/article.cfm?id=the-semantic-web.

BERNERS-LEE, T., MASINTER, L. y MCCAHILL, M. Uniform Resource Locators (URL). RFC 1738, Internet Engineering Task Force, 1994. http://ds.internic.net/rfc/rfc1738.txt.

BIKAKIS, N., GIANNOPOULOS, G., DALAMAGAS, T. y SELLIS, T. Integrating keywords and semantics on document annotation and search. En *Proceedings of the 2010 international conference on On the move to meaningful internet systems: Part II*, OTM'10, páginas 921–938. Springer-Verlag, Berlin, Heidelberg, 2010. ISBN 3-642-16948-1, 978-3-642-16948-9. http://portal.acm.org/citation.cfm?id=1926129.1926157.

BOIKO, B. *Content Management Bible*. John Wiley & Sons, Inc., New York, NY, USA, 2001. ISBN 076454862X.

BOLEY, H. y KIFER, M. RIF Basic Logic Dialect. W3C recommendation, W3C, 2010a. http://www.w3.org/TR/rif-bld/.

BOLEY, H. y KIFER, M. RIF Framework for Logic Dialects. W3C recommendation, W3C, 2010b. http://www.w3.org/TR/2010/REC-rif-fld-20100622/.

BRATSAS, C., DIMOU, A., IOANNIDIS, L., BAMIDIS, P. y ANTONIOU, I. Semantic CMS and Wikis as platforms for Linked Learning. En *Linked Learning 2012: 2nd International Workshop on Learning and Education with the Web of Data (LiLe2012)*. 2012. http://data.semanticweb.org/workshop/lile/2012/paper/26.

BRAY, T., HOLLANDER, D., LAYMAN, A. y TOBIN, R. Namespaces in XML 1.0. W3C recommendation, W3C, 2009. Published online on August 16th, 2006 at http://www.w3.org/TR/2009/REC-xml-names-20091208/.

BRAY, T., PAOLI, J., SPERBERG-MCQUEEN, C. M., MALER, E. y YERGEAU, F. Extensible Markup Language (XML) 1.0 (Fifth Edition). World Wide Web Consortium, Recommendation REC-xml-20081126, 2008. http://www.w3.org/TR/2008/REC-xml-20081126.

BRICKLEY, D. y GUHA, R. V. RDF Vocabulary Description Language 1.0: RDF Schema. Informe técnico, W3C, 2004. Availableonlineathttp://www.w3.org/TR/2004/REC-rdf-schema-20040210/.

BRY, F. y LAVINIA P ATRANJAN, P. Reactivity on the Web: Paradigms and Applications of the Language XChange. *J. of Web Engineering*, vol. 5, página 2006, 2005.

CASTELLS, P., BRAVO, C. y REDONDO, M. A. La Web Semántica. *Sistemas Interactivos y Colaborativos en la Web*, (39), páginas 195–212, 2003.

CHRIST, M., KRISHNAN, R., NAGIN, D. y GUNTHER, O. *Measuring Web portal utilization*, vol. 00, páginas 2647–2653. IEEE, 2002. `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=994220`.

CIMIANO, P. y VOLKER, J. Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. En *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)* (editado por A. Montoyo, R. M. noz y E. Metais), vol. 3513 de *Lecture Notes in Computer Science*, páginas 227–238. Springer, Alicante, Spain, 2005. `http://www.aifb.uni-karlsruhe.de/WBS/jvo/publications/Text2Onto_nldb_2005.pdf`.

CONSORTIUM, T. U. *The Unicode Standard: Worldwide Character Encoding. Version 1.0. Volumes 1 and 2*. 1991. ISBN 0-201-56788-1 (paperback, vol. 1), 0-201-60845-6 (paperback, vol. 2).

DEBRUIJN, J. RIF RDF and OWL Compatibility. W3C recommendation, W3C, 2010. `http://www.w3.org/TR/rif-rdf-owl/`.

DENNY, M. Ontology Tools Survey. Informe técnico, XML.com, 2004. `http://www.xml.com/lpt/a/2004/07/14/onto.html`.

DEROSE, S., MALER, E. y DANIEL, R. XML Pointer Language (XPointer) Version 1.0. Informe técnico, W3C, 2001. `http://www.w3.org/TR/2001/CR-xptr-20010911/`.

DUERST, M. y SUIGNARD, M. Internationalized Resource Identifiers (IRIs). RFC 3987, 2005. `http://rfc-ref.org/RFC-TEXTS/3987/index.html`.

FERNÁNDEZ-GARCÍA, N., BLÁZQUEZ-DEL-TORO, J., FISTEUS, J. y SÁNCHEZ-FERNÁNDEZ, L. *A Semantic Web Portal for Semantic Annotation and Search*, páginas 580–587. 2006. `http://dx.doi.org/10.1007/11893011_74`.

FONSECA, J. M. C., SUREDA, J. J. H. y ZABALA, P. A. R. Web semántica: Tecnologías y arquitectura. *Comunicaciones de Telefónica I+D*, (39), páginas 211–221, 2006. ISSN 1130-4693.

GIOVANNETTI, E., MARCHI, S., MONTEMAGNI, S. y BARTOLINI, R. Ontology Learning and Semantic Annotation: a Necessary Symbiosis. En *Proceedings of the Sixth International Language Resources and Evaluation*

(LREC'08) (editado por B. M. J. M. J. O. S. P. D. T. Nicoletta Calzolari (Conference Chair), Khalid Choukri). European Language Resources Association (ELRA), Marrakech, Morocco, 2008. ISBN 2-9517408-4-0. http://www.lrec-conf.org/proceedings/lrec2008/.

GOLBREICH, C. y WALLACE, E. K. OWL 2 Web Ontology Language: New Features and Rationale. World Wide Web Consortium, Working Draft WD-owl2-new-features-20081202, 2008. http://www.w3.org/TR/2008/WD-owl2-new-features-20081202.

GÓMEZ-PÉREZ, A., ANGELE, J., FERNÁNDEZ-LÓPEZ, M., CHRISTOPHIDES, V., STUTT, A. y SURE, Y. A survey on ontology tools. Deliverable 1.3, EU IST Project IST-2000-29243 OntoWeb, 2002. http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/OntoWeb_Del_1-3.pdf.

GÓMEZ-PÉREZ, A., FERNÁNDEZ-LÓPEZ, M. y CORCHO, O. *Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Advanced Information and Knowledge Processing. Springer, 1st edición, 2004. ISBN 1-85233-551-3.

GRUBER, T. R. A translation approach to portable ontology specifications. *Knowl. Acquis.*, vol. 5, páginas 199–220, 1993. ISSN 1042-8143. http://portal.acm.org/citation.cfm?id=173743.173747.

GUARINO, N. Understanding, building and using ontologies. *International Journal of Human-Computer Studies*, vol. 46(2-3), páginas 293–310, 1997. ISSN 1071-5819. http://portal.acm.org/citation.cfm?id=250543.

HALEVY, A., FRANKLIN, M. y MAIER, D. Principles of dataspace systems. En *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '06, páginas 1–9. ACM, New York, NY, USA, 2006. ISBN 1-59593-318-2. http://doi.acm.org/10.1145/1142351.1142352.

HANDSCHUH, S., STAAB, S. y STUDER, R. Leveraging Metadata Creation for the Semantic Web with CREAM. En *KI* (editado por A. Günter, R. Kruse y B. Neumann), vol. 2821 de *Lecture Notes in Computer Science*, páginas 19–33. Springer, 2003a. ISBN 3-540-20059-2. http://dblp.uni-trier.de/db/conf/ki/ki2003.html#HandschuhSS03.

HANDSCHUH, S., STAAB, S. y VOLZ, R. On Deep Annotation. En *Proceedings of the 12th International World Wide Web Conference, WWW 2003, Budapest, Hungary*. Budapest, Hungary, 2003b.

HAROLD BOLEY, M. K. A. P. A. P., GARY HALLMARK y REYNOLDS, D. RIF Core Dialect. 2010. http://www.w3.org/TR/rif-core/.

HAYDER, H. y SILVER, A. H. *WordPress 2.7 Complete*. Packt Publishing, 2009. ISBN 184719656X, 9781847196569.

HEATH, T. y BIZER, C. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 1st edición, 2011. ISBN 9781608454303. http://linkeddatabook.com/.

HITZLER, P., KROTZSCH, M., PARSIA, B., PATEL-SCHNEIDER, P. F. y RUDOLPH, S. OWL 2 Web Ontology Language Primer. W3C Recommendation, World Wide Web Consortium, 2009. http://www.w3.org/TR/owl2-primer/.

HORRIDGE, M. y PATEL-SCHNEIDER, P. F. OWL 2 Web Ontology Language: Manchester Syntax. Informe técnico, W3C, 2009. http://www.w3.org/TR/owl2-manchester-syntax/.

HORROCKS, I., PATEL-SCHNEIDER, P. F. y VAN HARMELEN, F. From SHIQ and RDF to OWL: The Making of a Web Ontology Language . *Journal of Web Semantics*, vol. 1(1), 2003.

KAHAN, J. y KOIVUNEN, M.-R. Annotea: an open RDF infrastructure for shared Web annotations. En *WWW '01: Proceedings of the 10th international conference on World Wide Web*, páginas 623–632. ACM Press, New York, NY, USA, 2001. ISBN 1581133480. http://dx.doi.org/10.1145/371920.372166.

KALYANPUR, A., PARSIA, B., HENDLER, J. y GOLBECK, J. SMORE: Semantic Markup, Ontology, and RDF Editor. 2005. http://www.mindswap.org/papers/SMORE.pdf.

KESSELMAN, J., ROBIE, J., CHAMPION, M., SHARPE, P., APPARAO, V. y WOOD, L. Document Object Model (DOM) Level 2 Traversal and Range Specification. W3C Recommendation, 2000. http://www.w3.org/TR/DOM-Level-2-Traversal-Range.

KHARE, R. Microformats: The Next (Small) Thing on the Semantic Web? *IEEE Internet Computing*, vol. 10, páginas 68–75, 2006. ISSN 1089-7801. http://doi.ieeecomputersociety.org/10.1109/MIC.2006.13.

KHONDOKER, R. y MÜLLER, P. Comparing Ontology Development Tools Based on an Online Survey. *World Congress on Engineering*, 2010.

KIFER, M. y BOLEY, H. RIF Overview. 2010. http://www.w3.org/TR/2010/NOTE-rif-overview-20100622/.

KRAMER, J. *Joomla! Start to Finish: How to Plan, Execute, and Maintain Your Web Site*. Wrox Press Ltd., Birmingham, UK, UK, 2010. ISBN 047057089X, 9780470570890.

LASSILA, O. y MCGUINNESS, D. L. The Role of Frame-Based Representation on the Semantic Web. Knowledge Systems Laboratory Report KSL-01-02, Stanford University, 2001. Also appeared as Linköping Electronic Articles in Computer and Information Science, Vol. 6 (2001), No. 005, Linköping University.

LE, D. M. y LAU, L. M. S. An Open Architecture for Ontology-Enabled Content Management Systems: A Case Study in Managing Learning Objects. En *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, OTM Confederated International Conferences, CoopIS, DOA, GADA, and ODBASE 2006, Montpellier, France, October 29 - November 3, 2006. Proceedings, Part I* (editado por R. Meersman y Z. Tari), vol. 4275 de *Lecture Notes in Computer Science*, páginas 772–790. Springer, 2006. ISBN 3-540-48287-3.

LUHN, H. P. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, vol. 2, páginas 159–165, 1958. ISSN 0018-8646. `http://dx.doi.org/10.1147/rd.22.0159`.

MAEDCHE, A., MOTIK, B., STOJANOVIC, L., STUDER, R. y VOLZ, R. Ontologies for Enterprise Knowledge Management. *IEEE Intelligent Systems*, vol. 18(2), páginas 26–33, 2003. `http://csdl2.computer.org/dl/mags/ex/2003/02/x2026.pdf`.

MANOLA, F. y MILLER, E. RDF Primer. W3C recommendation, W3C, 2004. Published online on February 10th, 2004 at `http://www.w3.org/TR/2004/REC-rdf-primer-20040210/`.

MARTIN-RECUERDA, F., HARTH, A., DECKER, S., ZHDANOVA, A., DING, Y., STOLLBERG, M. y ARROYO, S. D2.1 Report on Requirements Analysis and State of the Art (WP2 - Ontology Management). Informe Técnico 1.0, Data, Information, and Process Integration with Semantic Web Services - Project DIP, 2004. `http://dip.semanticweb.org/documents/DIP-D21-v1_Public.pdf`.

MCGUINNESS, D. L. y VAN HARMELEN, F. OWL Web Ontology Language Overview. W3c recommendation, World Wide Web Consortium, 2004. `http://www.w3.org/TR/2004/REC-owl-features-20040210/`.

MERCER, D. *Building Powerful and Robust Websites with Drupal 6: Build your own professional blog, forum, portal or community website with Drupal 6*. Packt Publishing, 2008. ISBN 1847192971.

MILES, A. y BECHHOFER, S. SKOS Simple Knowledge Organization System Reference. World Wide Web Consortium, Working Draft WD-skos-reference-20080829, 2008. `http://www.w3.org/TR/2008/WD-skos-reference-20080829`.

MOTIK, B., GRAU, B. C., HORROCKS, I., WU, Z., FOKOUE, A. y LUTZ, C. OWL 2 Web Ontology Language: Profiles. W3c recommendation, W3C, 2008a. http://www.w3.org/TR/2008/WD-owl2-profiles-20081008/.

MOTIK, B., PATEL-SCHNEIDER, P. F. y GRAU, B. C. OWL 2 Web Ontology Language: Direct Semantics. World Wide Web Consortium, Working Draft WD-owl2-semantics-20081202, 2008b. http://www.w3.org/TR/2008/WD-owl2-semantics-20081202.

MOTIK, B., PATEL-SCHNEIDER, P. F., PARSIA, B., BOCK, C., FOKOUE, A., HAASE, P., HOEKSTRA, R., HORROCKS, I., RUTTENBERG, A., SATTLER, U. y SMITH, M. OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. W3c recommendation, W3C, 2008c. (to be published, may be superseded).

NAVARRO-GALINDO, J. L. y SAMOS, J. Integración de Información en la Web Semántica. En *Trabajo de Investigación DEA*. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada, Granada, Spain, 2007a.

NAVARRO-GALINDO, J. L. y SAMOS, J. Una Panorámica Actual de Software para Trabajar con Ontologías. En *I Simposio en Desarrollo Software (SDS07)*, páginas 101–116. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada, Granada, Spain, 2007b. ISBN 978-84-96856-17-2.

NAVARRO-GALINDO, J. L. y SAMOS, J. Flexible range semantic annotations based on RDFa. En *27th British National Conference on Databases: Data security and security data*, páginas 122–126. Springer-Verlag, Dundee, United Kingdom, 2010a. ISBN 978-3-642-25703-2. http://dx.doi.org/10.1007/978-3-642-25704-9.

NAVARRO-GALINDO, J. L. y SAMOS, J. Manual and Automatic Semantic Annotation of Web Documents: The FLERSA Tool. En *12th International Conference on Information Integration and Web-based Applications and Services (iiWAS 2010)*, vol. 1, páginas 540–547. ACM, Paris, France, 2010b. ISBN 978-1-4503-0421-4.

NAVARRO-GALINDO, J. L. y SAMOS, J. FLERSA: Soporte a la Definición de Anotaciones y Búsquedas Semánticas en un CMS. En *Actas de las XVI Jornadas de Ingeniería del Software y Bases de Datos*, páginas 101–114. Universidade da Coruña, A Coruña, Spain, 2011. ISBN 978-84-9749-486-1. http://www.sistedes.es/jornadas2011/jisbd.htm.

NAVARRO-GALINDO, J. L. y SAMOS, J. The FLERSA tool: Adding Semantics to a Web Content Management System. *International Journal of Web*

Information Systems, vol. 8(1), páginas 73–126, 2012. ISSN 1744-0084. http://dx.doi.org/10.1108/17440081211222609.

NOY, N. F. y MCGUINNESS, D. L. Ontology Development 101: A Guide to Creating Your First Ontology. Informe técnico, Stanford Knowledge Systems Laboratory and Stanford Medical Informatics, 2001. http://www.ksl.stanford.edu/KSL_Abstracts/KSL-01-05.html.

OLDAKOWSKI, R., BIZER, C. y WESTPHAL, D. RAP: RDF API for PHP. En *In proc. International Workshop on Interpreted Languages*. MIT Press, 2004.

PATEL-SCHNEIDER, P., HAYES, P. y HORROCKS, I. Web Ontlogy Language (OWL) Abstract Syntax and Semantics. 2004. http://www.w3.org/TR/2004/REC-owl-semantics-20040210/.

PEACOCK, M. *Building Websites with TYPO3: A practical guide to getting your TYPO3 website up and running fast*. Packt Publishing, 2007. ISBN 1847191118, 9781847191113.

PEROJO, K. R. y LEÓN, R. R. Web semántica: un nuevo enfoque para la organización y recuperación de información en la web. *ACIMED*, vol. 13(6), páginas 0–0, 2005. ISSN 1024-9435.

POLLERES, A., BOLEY, H. y KIFER, M. RIF Datatypes and Built-Ins 1.0. W3C recommendation, W3C, 2010. http://www.w3.org/TR/rif-dtb/.

POPOV, B., KIRYAKOV, A., OGNYANOFF, D., MANOV, D. y KIRILOV, A. KIM - a semantic platform for information extraction and retrieval. *Nat. Lang. Eng.*, vol. 10, páginas 375–392, 2004. ISSN 1351-3249. http://dx.doi.org/10.1017/S135132490400347X.

PRUD'HOMMEAUX, E. y SEABORNE, A. SPARQL Query Language for RDF. W3c recommendation, 2008. http://www.w3.org/TR/rdf-sparql-query/.

QUINT, V. y VATTON, I. An introduction to Amaya. *World Wide Web J.*, vol. 2, páginas 39–46, 1997. ISSN 1085-2301. http://portal.acm.org/citation.cfm?id=275062.275068.

DE SAINTE MARIE, C., PASCHKE, A. y HALLMARK, G. RIF Production Rule Dialect. World Wide Web Consortium, Working Draft WD-rif-prd-20081218, 2010. http://www.w3.org/TR/2008/WD-rif-prd-20081218.

SCHNEIDER, M. OWL 2 Web Ontology Language: RDF-Based Semantics. Informe técnico, 2009. http://www.w3.org/TR/2009/REC-owl2-rdf-based-semantics-20091027/.

SHETH, A., BERTRAM, C., AVANT, D., HAMMOND, B., KOCHUT, K. y WARKE, Y. Managing Semantic Content for the Web. *IEEE Internet Computing*, vol. 6, páginas 80–87, 2002. ISSN 1089-7801. http://portal.acm.org/citation.cfm?id=613355.613729.

SMITH, B. Ontology. an introduction. *In: Floridi, L. (Ed.),Blackwell Guide to the Philosophy of Computing and Information, Blackwell, Oxford*, páginas 155–166, 2003.

SOWA, J. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks/Cole, Pacific Grove, CA, 2000. ISBN 978-0-534-94965-5.

SPARCK JONES, K. *A statistical interpretation of term specificity and its application in retrieval*, páginas 132–142. Taylor Graham Publishing, London, UK, 1988. ISBN 0-947568-21-2. http://portal.acm.org/citation.cfm?id=106765.106782.

STENBACK, J., HÉGARET, P. L. y HORS, A. L. Document Object Model (DOM) Level 2 HTML Specification. W3C Recommendation, 2003. http://www.w3.org/TR/2003/REC-DOM-Level-2-HTML-20030109.

STUDER, R., BENJAMINS, V. R. y FENSEL, D. Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering*, vol. 25(1-2), páginas 161–197, 1998. http://dx.doi.org/10.1016/S0169-023X(97)00056-6.

TSAI, T.-M., YU, H.-K., LIAO, P.-Y. y SHIH, H.-T. Semantic Modeling among Web Services Interfaces for Services Integration-SOTA(Smart Office Task Automation) platform. En *Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, DEXA '03. IEEE Computer Society, Washington, DC, USA, 2003. ISBN 0-7695-1993-8. http://portal.acm.org/citation.cfm?id=942790.942968.

TUMMARELLO, G. y MORBIDONI, C. The DBin platform: A complete environment for Semantic Web Communities. *Web Semant.*, vol. 6, páginas 257–265, 2008. ISSN 1570-8268. http://portal.acm.org/citation.cfm?id=1464505.1464599.

UREN, V., CIMIANO, P., IRIA, J., HANDSCHUH, S., VARGAS-VERA, M., MOTTA, E. y CIRAVEGNA, F. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semant.*, vol. 4, páginas 14–28, 2006. ISSN 1570-8268. http://dx.doi.org/10.1016/j.websem.2005.10.002.

VARGAS-VERA, M., MOTTA, E., DOMINGUE, J., LANZONI, M., STUTT, A. y CIRAVEGNA, F. MnM: Ontology Driven Semi-Automatic and Automatic

Support for Semantic Markup. 2002. `http://citeseer.ist.psu.edu/article/vargas-vera02mnm.html`.

ZACHARIAS, V. y BRAUN, S. SOBOLEO - Social Bookmarking and Lightweight Engineering of Ontologies. En *Proc. WWW 2007 Workshop on Social and Collaborative Construction of Structured Knowledge*. Banff, Canada, 2007.

Lista de Acrónimos

AJAX ..... *Asincronous Javascript And XML*, Javascript y XML asíncrono

API ..... *Application Programming Interface*, Interfaz de Programación de Aplicaciones

CERN ..... *Conseil Européen pour la Recherche Nucléaire*

CMS ..... *Content Management System*, Sistema de Gestión de Contenido

DCMI ..... *Dublin Core Metadata Initiative*, Iniciativa de Metadatos Dublin Core

DTD ..... *Document Type Definition*

ECMS ..... *Enterprise Content Management System*

IRI ..... *Internationalized Resource Identifiers*, Identificador de Recursos Internacionales

MIME ..... *Multipurpose Internet Mail Extensions*

MVC ..... *Model View Controller*, Modelo-Vista-Controlador

NS ..... *NameSpaces*, Espacios de Nombres

OKBC ..... *Open Knowledge Base Connectivity*, Conectividad Abierta de la Base de Conocimiento

OWL ..... *Web Ontology Language*, Lenguaje de Ontologías Web

RDF ..... *Resource Description Framework*, Marco de Descripción de Recursos

RDFS ..... *RDF Schema*

RIF ..... *Rule Interchange Format*, Formato de Intercambio de Reglas

RIF-BLD ..... *RIF Basic Logic Dialect*, Dialecto RIF de Lógica Básica

RIF-DTB ..... *RIF Datatypes and Built-ins*, Tipos de Dato y Empotrados RIF

RIF-FLD ..... *RIF Framework for Logic Dialects*, Marco de trabajo RIF para Dialectos Lógicos

RIF-PRD ..... *RIF Production Rule Dialect*, Dialecto RIF de Producción de Reglas

S-CMS ..... *Semantic Content Management System*, Sistema Semántico de Gestión de Contenido

SHOE ..... *Simple HTML Ontology Extensions*

SKOS ..... *Simple Knowledge Organization System*, Sistema de Organización Simple de Conocimiento

SPARQL ..... *SPARQL Protocol and RDF Query Language*, Protocolo SPARQL y Lenguaje de Consulta RDF

SWRL ..... *Semantic Web Rule Language*, Lenguaje de Reglas de la Web Semántica

UNA ..... *Unique Name Assumption*

URI ..... *Uniform Resource Identifier*, Identificador Uniforme de Recursos

URL ..... *Uniform Resource Locator*, Localizador Uniforme de Recursos

W3C ..... *World Wide Web Consortium*

WCMS ..... *Web Content Management System*, Sistema de Gestión de Contenido Web

WWW ..... *World Wide Web*

WYSIWYG ... *What You See Is What You Get*

XML ..... *eXtensible Markup Languaje*, Lenguaje de Marcas Extensible

XPATH ..... *XML Path Language*

–¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.

Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes

–Buena está – dijo Sancho –; fírmela vuestra merced.
–No es menester firmarla – dijo Don Quijote–,
sino solamente poner mi rúbrica.

Primera parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes