

Aplicación de tecnologías de la Web Semántica a la gestión de información financiera y económica

J. Bravo, C. Carranza,
P. Castells, J.M. Fuentes,
M. Rico

Universidad Autónoma de Madrid
Ctra. de Colmenar Viejo km. 15
28049 Madrid
Spain

{javier.bravo, cesar.carranza,
pablo.castells, chema.fuentes,
mariano.rico} @uam.es

J. L. Alonso, B. Foncillas,
J.M Rodríguez

Tecnología Información y Finanzas
C/ Españolto 19
28010 Madrid
Spain

{jalonso, bfoncillas,
jmrodriguez} @afi.es

R. Lara

Universität Innsbruck
Techniker Strasse 13
6020 Innsbruck
Austria

ruben.lara@uibk.ac.at

Resumen

Se describe la experiencia adquirida como resultado de la creación de una aplicación web, con técnicas de la Web Semántica, para la gestión de la información financiera y económica de una destacada empresa del sector. El trabajo desarrollado proporciona una ontología para el dominio financiero y económico, búsqueda semántica, así como la visualización y navegación por los contenidos en las aplicaciones web usadas por los clientes de dicha empresa.

Las técnicas de la Web Semántica plantean nuevos retos en cuanto al diseño del interfaz de usuario, mostrándose las soluciones aportadas.

Palabras clave

Web Semántica, ontologías, diseño de Interfaces de Usuario, Recuperación de Información, Lenguajes de Consulta, Visualización, Economía y Finanzas.

1. Introducción

Aunque han mejorado mucho los buscadores desde sus orígenes, es bien conocida la frustración que genera una búsqueda en Internet del estilo “valores IBEX35 que han subido hoy”. Incluso Google, la empresa que procesa el 40% de las búsquedas realizadas en Internet, y pese a sus 10.000 ordenadores, no “entenderá” la pregunta aunque devuelva 439 referencias. Ninguna de las diez primeras referencias responderá a nuestra pregunta.

El dominio de la información financiera y económica es conceptualmente muy amplio, complejo, voluminoso, y con un valor muy alto. Cada día se genera un enorme volumen de información, haciendo que no sea procesable por una única persona. Se necesitan mecanismos eficientes de clasificación, filtrado, búsqueda y navegación para los consumidores de dicha información con el fin de que puedan acceder a los contenidos más relevantes según el perfil de cada usuario, aportando así un alto valor añadido a la información. Esta necesidad explica que la

comunidad financiera sea uno de los mayores consumidores de tecnologías de la información.

Las tecnologías emergentes de la Web Semántica están generando nuevas formas de representación y distribución de la información, y prometen soluciones al problema planteado inicialmente. Estas soluciones pasan por el uso de agentes personales que “entiendan” nuestras preguntas y busquen las respuestas por la red. Esta nueva red, paralela a la tradicional “para humanos”, es la que se ha venido a denominar Web Semántica, en la que la información está estructurada y dispuesta para su consumo por parte de los agentes mencionados. Naturalmente no son redes desacopladas, sino que la web tradicional se enriquecerá por la existencia de la nueva. Por tanto, el primer paso consiste en preparar semánticamente la información, bien la nueva, generada día a día como en nuestro caso, bien la tradicional de las páginas web. El primer caso es más sencillo que el segundo, porque la clasificación semántica de la información es realizada por humanos en el momento en que se produce esa información. En el segundo caso, el volumen de la web actual obliga a realizar esta tarea de forma automática y, por tanto, más propensa a errores.

En este artículo se propone una plataforma basada en ontologías para la gestión de los contenidos financieros y económicos. Esta plataforma proporciona la siguiente funcionalidad: búsqueda, visualización y navegación basadas en semántica.

Las empresas dedicadas a la venta de información de este dominio, como veremos en la sección 2, se pueden beneficiar de las ventajas aportadas por las recientes técnicas de la Web Semántica, como veremos en la sección 3. La utilización de estas técnicas, que forman la base del proyecto descrito en este artículo, se verán en la sección 4. Éstas permiten la estructuración del conocimiento del dominio así como la búsqueda de información de una manera mucho más refinada que las técnicas tradicionales. La sección 5 mostrará las conclusiones de nuestro proyecto.

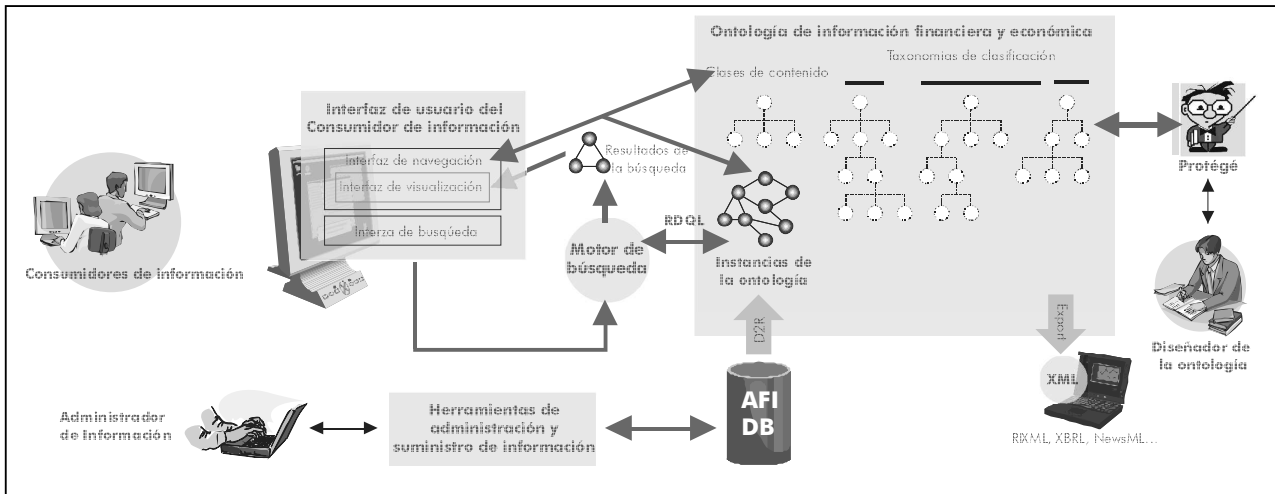


Figura 1. Módulos de la aplicación Aniceto.

2. AFI, proveedores de información financiera y económica

EL Grupo Analistas Financieros Internacionales (AFI) es una compañía española que genera información económica y financiera de alta calidad diariamente. Sus expertos generan notas de investigación, análisis de mercado económicos y financieros, información sobre ferias comerciales, informes por sectores, recomendaciones, noticias, manuales, etc. Tecnología, Información y Finanzas (TIF) es la empresa creada por AFI para aportar las tecnologías de la información necesarias. AFI ha creado portales web personalizados en función de los intereses de cada cliente. Para caracterizar a cada cliente, éste elige los contenidos que le interesan seleccionándolos de una carta de categorías.

Esta carta, que incluye análisis de mercados, ofertas de importación/exportación para PYMEs de comercio internacional, análisis bancario, América Latina, sector eléctrico y muchas otras categorías, es usada diariamente por Bancos, Cajas de Ahorros, y todo tipo de grandes entidades, para su toma de decisiones. También es usada por PYMEs para, por ejemplo, sus actuaciones en comercio exterior.

Algunos clientes son redistribuidores a su vez de esta información en medios impresos y digitales referentes de la economía española. Por tanto, la adecuación de esta información a las necesidades de los clientes y la facilidad de integración, procesamiento y distribución de la misma es de gran importancia para la gestión eficaz de la información económica nacional.

3. Aplicación de tecnologías de la Web Semántica a la gestión de información

3.1 Web Semántica

La Web actual está concebida para ser utilizada e interpretada en su mayor parte por seres humanos. Por su parte, la Web

Semántica [3] no es una Web separada, sino una extensión de la actual, en la que la información cuenta con una representación bien definida de su significado. Así se permite también a las aplicaciones una mayor “comprensión” de los datos, ampliando las posibilidades de gestión de esta información. Estas metas se reflejan en el desarrollo de nuevas técnicas que se basan en la introducción de conocimiento semántico explícito que describen y estructuran la información y los servicios disponibles de forma susceptible de ser procesada automáticamente por una aplicación. Uno de los ejes principales de esta visión es la noción de Ontología, herramienta clave para representar la semántica del dominio.

Heredada de la Inteligencia Artificial, una ontología es “una especificación explícita formal de una conceptualización común” [4], es decir, proporciona un modelo explícito obtenido por consenso descrito en un lenguaje que contiene a los conceptos, propiedades y relaciones más relevantes de un dominio. Hay diversos lenguajes formales para definir ontologías. Por orden histórico serían SHOE (1997), y desde la aparición de XML en 1998: RDF (1999), RDF Schema (RDFS), DAML, OIL, DAML+OIL y OWL.

Las aplicaciones usan las ontologías como vocabulario común usado por las partes implicadas. Esto obliga a que las ontologías sean accesibles a través de la red haciendo de las aplicaciones verdaderos integradores de información distribuida. Esta idea de información distribuida es otro de los pilares de la la Web Semántica. Si se pretenden “entender” los distintos puntos de vista, las distintas propuestas, los diferentes vocabularios deberán exponerse públicamente y por tanto ser accesibles por la red.

La infraestructura utilizada actualmente por TIF para administrar la información financiera y económica, creada conforme a las necesidades de su negocio, se presta a ser utilizada usando las tecnologías de la Web Semántica. Con ello se ha pretendido lograr una gestión y producción más eficiente de sus contenidos financieros y económicos.

3.2 Una propuesta: Proyecto ANICETO

A principios de 2003 el Ministerio de Ciencia y Tecnología aprobó el proyecto PROFIT presentado por el grupo de Web Semántica de la Universidad Autónoma de Madrid, en colaboración con TIF y el grupo Next Web Generation de la Universidad de Innsbruck. Este proyecto recibió el nombre de Aniceto y ha tenido como objetivos:

- Definir una ontología para el dominio financiero y económico
- Desarrollar gestores de generación de contenido para mostrar los datos de la ontología definida.
- Desarrollar una aplicación web que permita una navegación semántica en los portales web generados “a medida” para los diversos clientes.
- Soporte de búsqueda semántica en los contenidos que aumente considerablemente la precisión de la búsqueda en comparación con las búsquedas tradicionales.
- Inclusión de perfiles de usuario, que permitan una personalización de los contenidos
- Que no interfiera con el sistema en explotación aunque las fuentes de información sean comunes.

4. Arquitectura de ANICETO

Aniceto se compone de los módulos representados en la Figura 1.

El modelo de datos es un modelo basado en ontologías (MDBO) de dominio público llamado Jena, implementado en Java por HP Labs [6]. Este modelo de datos permite manipular las ontologías y tener los datos en memoria o en base de datos. También permite consultas avanzadas al modelo en el lenguaje de consultas RDQL.

El módulo del buscador permite la conexión de nuestro modelo de datos con el servidor Web. El usuario interactúa con el sistema a través de un navegador convencional, realizando consultas al sistema seleccionando un contenido de la ontología (Noticia, Indicador, Informe, etc.). Como respuesta, el sistema le muestra los atributos del contenido seleccionado para que el usuario rellene los datos que le resulten más relevantes. Adicionalmente también puede seleccionar la categoría en la que está clasificada el contenido. Una vez formulada la consulta se envía al sistema y éste devuelve los resultados adecuados. El resultado devuelto puede que contenga instancias de distintos tipos, por lo que cada vez que se seleccione una instancia, es el módulo de visualización el encargado de mostrarla adecuadamente.

El sistema utiliza servlets y páginas JSP siguiendo el paradigma de diseño MVC. La base de datos de AFI es una conocida marca comercial y la base de datos del MDBO es MySQL.

4.1 Definición y gestión de ontologías

Como hemos visto anteriormente, una ontología es una puesta en común de clases (conceptos) con atributos y relaciones entre clases. En el caso de Aniceto, la primera versión de la ontología financiera y económica de AFI fue creada a partir del modelo de base de datos que existía. En versiones posteriores fue enriquecida con nuevos conceptos, atributos y relaciones, extraídos a partir del conocimiento adquirido en las reuniones con los expertos de AFI.

Para la creación y el modelado de la ontología se ha utilizado la herramienta de edición de ontologías Protege [9] (ver Figura 2).

El lenguaje de ontologías escogido fue RDFS y RDF. Se utilizó RDFS para la definición de los conceptos (clases), atributos y relaciones de la ontología, y RDF para las instancias. Aunque no sea un lenguaje tan expresivo como otros lenguajes más modernos como DAML, OIL, o OWL, está más asentado y es el más ampliamente soportado por este tipo de herramientas.

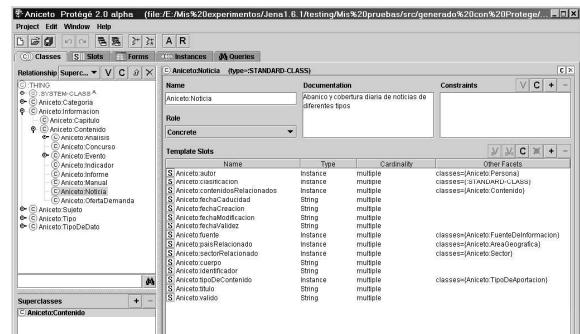


Figura 2. Editor de ontologías Protégé.

Una de las hipótesis a comprobar en este proyecto era saber si la expresividad proporcionada por RDFS/RDF es suficiente para el dominio en cuestión. La experiencia adquirida en el transcurso del proyecto nos ha permitido concluir que nuestra hipótesis era correcta pues en ningún momento hemos echado en falta mayor capacidad de conceptualización en este dominio.

La ontología final tiene como clases básicas a *Contenido*, *Sujeto* y *CategoriaAniceto* como muestra la Figura 3. El resto de las clases son clases derivadas (o hijas) de estas. El concepto Contenido (y sus 16 clases derivadas) denota la información especializada generada por AFI: Noticia, Análisis técnico, Informe, Evento, etc. El concepto Sujeto (y sus 19 clases derivadas) describe los actores, entidades y objetos involucrados en el dominio, por ejemplo: Área Geográfica, Banco, Recinto Ferial, Persona, etc. El concepto CategoríaAniceto (y sus 105 clases derivadas) es una taxonomía (clases sin relaciones con otras clases fuera de la taxonomía) para clasificar la información. Describe a primer nivel 10 clases (Banca, Análisis de Mercados, Economía, Empresas, Fiscalidad Empresarial, entre otras) y el resto de clases son “especializaciones” de estas clases del primer nivel. Las clases de esta taxonomía no se pueden instanciar.

4.1.1 Estándares financieros

En el dominio de las finanzas y la economía se usa desde hace tiempo XML como formato de intercambio de información. Se ha extendido el uso de algunos estándares XML para representar estos contenidos, como por ejemplo FpML o NewsML entre otros. AFI vende noticias también en estos formatos electrónicos, por lo que la importación y exportación de información en estos estándares también era un requisito para la correcta integración de Aniceto.

La fuerte estructuración de la información gracias a las técnicas de la Web Semántica se adecua perfectamente a este requisito, por lo

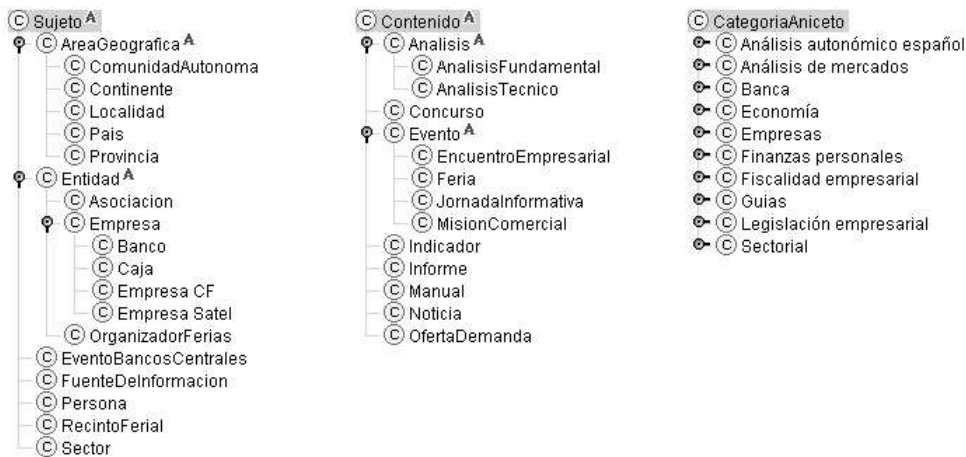


Figura 3. Elementos principales de la ontología del proyecto Aniceto.

que la exportación de información en cualquiera de los formatos XML es una tarea sencilla que no ha supuesto ningún problema técnico. Hay que destacar que las ontologías (taxonomías) usadas por estos estándares, como en el caso de la clasificación de las noticias, ha obligado a hacer una tabla de conversión entre nuestra clasificación, adaptada a las particularidades de AFI, y la del estándar. Esta tabla de conversión también es de dominio público.

4.2 Modelos de datos basados en ontologías (MDBO)

El uso de modelos de datos orientados a ontologías modificará el paradigma de diseño tradicional de aplicaciones. Estos modelos permiten la manipulación de ontologías a través de un API. Su nivel de abstracción es tal, que ocultan completamente la implementación física del modelo, esto es, ocultan el detalle de si los datos del modelo se encuentran en memoria, en un fichero, o en una BD. Más aún, en el caso de una implementación física sobre BD, los detalles de cómo se reparte la información por tablas, índices, y demás parámetros críticos del diseño en BD tradicionales quedan ocultos.

Este recubrimiento rompe el paradigma clásico de los tres niveles pues el nivel de BD queda oculto por el nivel de modelo lógico como muestra la figura 4. Por modelo lógico, queremos indicar la parte de la aplicación en la que se usa la información de la BD, la procesa, y la envía a la aplicación en el formato adecuado como muestra la Figura 4.

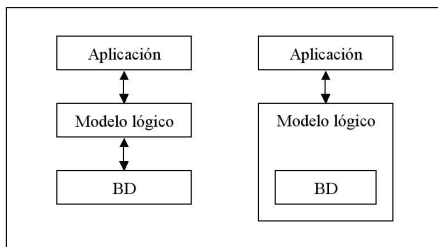


Figura 4. Cambio en el paradigma de tres niveles. A la izquierda el modelo tradicional. A la derecha el MDBO

Este cambio en el paradigma de diseño de aplicaciones supone un enorme reto, pues permite automatizar la implementación sobre BD del modelo lógico. La tarea de mantener coherente el modelo lógico y el modelo en BD ha sido tradicionalmente muy costosa, por lo que liberarse de esta tarea supone una importante ventaja.

El reto no sólo reside en su eficiencia, sino también en su flexibilidad. Es en este aspecto de flexibilidad donde la Intefaz de Usuario (IU) se vuelve decisiva: una correcta representación de la ontología y de los cambios que se desean realizar en ella permitiría la flexibilidad requerida para esta tarea. Este objetivo debe conseguirse para que los modelos ontológicos sean lo bastante flexibles como para resultar útiles.

Creemos que el MDBO aporta un nuevo punto de vista acerca de la información disponible en el modelo, siendo esta mucho más explícita que en BD, y con un modelo lógico más sencillo pues ahorra el mantenimiento de la consistencia con la BD. Esto supone una menor complejidad lógica y, por tanto, un menor tiempo de desarrollo de aplicaciones que usen este modelo.

En nuestro caso, el modelo MDBO consistirá en un grafo de triples (ver Figura 5) que se podría representar siguiendo el paradigma tradicional de BD como una única tabla de tres campos (sujeto, predicado, valor) en el que cualquier celda puede ser de un tipo complejo. Creemos que este es un hecho crucial que lo diferencia de las BD tradicionales donde los tipos están fijados (y varían entre distintos fabricantes de BDs) y que tiene una visualización inmediata que facilita su comprensión por parte del diseñador de la aplicación.

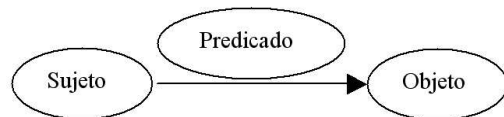


Figura 5. Modelo de triples.

Estos MDBOs se convierten en prácticos modelos genéricos o metamodelos (modelos de modelos) con una lógica común

Figura 6. Portal Aniceto. Búsqueda semántica. En este ejemplo se buscan Ferias (seleccionado en el menu desplegable Contenido) que estén en cualquier clasificación.

independiente del modelo concreto (representado por una o varias ontologías) que se esté usando en cada aplicación.

4.3 Generación y conversión de contenidos

AFI dispone de expertos de todas las áreas de la economía y las finanzas que cada día generan información económica y financiera y la almacenan en sus sistemas. Sus aplicaciones se encargan de hacerla llegar a sus clientes, bien de forma activa (envío de esta información en múltiples formatos electrónicos), bien de forma pasiva mediante la preparación de portales web.

Al ser este proyecto una extensión de un sistema en explotación, se disponía de un gran conjunto de datos. Estos datos se encontraban en una base de datos tradicional que hubo que migrar a RDF. Para esta tarea se empleó la herramienta de dominio público D2R [7]. Esta herramienta realiza una conexión JDBC (o ODBC) con la base de datos tradicional y, usando un fichero XML donde se define cómo mapear los campos de un ResultSet (resultado de una sentencia SQL) con los elementos de la ontología, genera las instancias en un fichero RDF. Este fichero es cargado en nuestro MDBO y añadido a su base de datos.

Esta tarea conlleva típicamente a la creación de un fichero de mapeo XML al menos por cada clase (concepto) de la ontología. Al tener que obtener todos los atributos de esta clase, podemos saber cómo de complejo es el concepto comparado con su implementación sobre la base de datos tradicional por la complejidad de la sentencia SQL. Cuanto más compleja es la sentencia SQL, más “enmascarado” es el concepto. Hemos comprobado que muchos conceptos de nuestra ontología se encontraban muy escondidos en la base de datos. Cuanto más compleja sea la base de datos, más conceptos escondidos puede tener y más útil puede resultar hacerlos explícitos mediante el uso de ontologías.

Un reto pendiente es la modificación de la ontología, pues implica reestructurar los datos antiguos para adaptarlos a la nueva ontología. Aunque es un tema aún sin resolver completamente, ya existen herramientas que facilitan, y en algunos casos

automatizan, la reestructuración de la información. Aún así no es un problema exclusivo de los MDBOs, pues en una aplicación tradicional, la reestructuración o rediseño de la BD conlleva la re-implimentación del modelo de datos así como la reconversión de los datos de la BD.

4.4 Módulo de búsqueda

El módulo de búsqueda permite al usuario seleccionar los conceptos de la ontología, para que una vez seleccionado se le muestren los campos por los que desea realizar la búsqueda. El aspecto del interfaz se muestra en la Figura 6.

La búsqueda semántica se diferencia de la búsqueda tradicional en que la primera no trata de encontrar un literal entre toda la información del sistema, sino que ofrece al usuario la posibilidad de explicar qué significa ese literal. Por ejemplo, si el usuario está interesado en localizar ferias de calzado en Alicante, a diferencia del sistema tradicional en el que introduciría simplemente el literal "ferias calzado Alicante", con el buscador semántico seleccionará el contenido Feria, se le mostrarán los campos (atributos) de ese concepto, y rellenará los campos pertinentes. Adicionalmente podrá indicar la categoría en la que está clasificado el contenido aunque por defecto se selecciona "Cualquiera". El módulo de búsqueda generará la sentencia RDQL oportuna. RDQL es el lenguaje de consultas de Jena para buscar instancias en el modelo RDF. En este ejemplo, seleccionando Feria como Contenido la consulta tendrá un aspecto parecido a lo siguiente:

```
SELECT ?x
WHERE (?x <type> Feria),
      (?x <sector> Calzado),
      (?x <localizacion> "Alicante")
```

Aunque esta sentencia nunca es visualizada por el usuario, sí hemos considerado la posibilidad de mostrar su equivalente en lenguaje natural, considerando que puede ser útil especialmente en consultas complejas. Esta sentencia será enviada al MDBO;

éste devolverá el conjunto de instancias RDF cuyo tipo sea FERIA (serán objetos de la clase FERIA o cualquiera de sus clases derivadas), su atributo *sector* sea Calzado y su atributo *localización* sea "Alicante". El atributo *sector* sería de tipo Calzado y el atributo *localización* sería de tipo Literal.

Siempre podremos realizar la consulta tradicional si como Contenido seleccionamos "Cualquiera" y como Clasificación seleccionamos Categoría (la clase base de las categorías) para indicar cualquier categoría. En este caso aparecerá únicamente un campo, permitiendo al usuario introducir el literal deseado. Si el valor introducido fuese "Alicante", ésta búsqueda generaría una sentencia RDQL como esta:

```
SELECT ?x
WHERE (?x, ?y, ?z)
AND ?z=~ ¡Alicante!
```

En este caso el sistema devolvería todos los objetos en los que alguno de sus atributos contenga el literal "Alicante".

Así, podemos ver claramente que la búsqueda tradicional es un caso particular de la búsqueda semántica.

Los autores han diseñado formas gráficas de generación de consultas RDQL, para refinar aún más la búsqueda. En el primer ejemplo mostrábamos Localidad como un literal, pero bien pudiera ser un concepto más de la ontología. En este nuevo ejemplo, Localización podría tener un campo perteneceA para indicar términos territoriales más amplios. La Figura 6 muestra el interfaz de usuario para este ejemplo. El signo + al lado del campo indica tipo complejo o elemento de lista.

4.4.1 Limitaciones de RDQL

Con RDQL sólo se pueden hacer consultas sobre elementos explícitamente presentes en el modelo de datos. No realiza ningún tipo de inferencia para desvelar relaciones implícitas entre los elementos del modelo. Un ejemplo sencillo puede ser la relación subclassOf. Esta relación se da entre las clases padre e hija, de manera que la clase hija tiene el atributo subclassOf con el valor de la clase padre. Pues bien, si tenemos una jerarquía de este estilo, en el que, siguiendo el caso descrito, tenemos una clase "abuela", padre de la clase padre, la relación abuela-nieta no será explícita. En el caso de nuestra ontología, la clase FERIA es subclassOf Evento, y esta a su vez es subclassOf Contenido. Si construimos una sentencia RDQL en la que se busquen Contenidos relacionados con Ferias, no se encontrará nada, pues no hay relación explícita entre Contenido y FERIA.

Afortunadamente, Jena dispone de un motor de inferencia que permite hacer explícitas ese tipo de relaciones. Esto implica un moderado coste, pues en nuestro caso todas y cada una de las instancias tendrán atributos subclassOf a todas las clases superiores. La ventaja será que se podrán realizar consultas más ricas aún, y más rápidamente que con otros lenguajes que generan las relaciones en el momento de la búsqueda, como SeRQL (de Sesame [8]).

El motor de inferencia de Jena es altamente configurable, siendo de particular interés su sistema basado en reglas.

4.5 Visualización y personalización

El resultado de una búsqueda es una lista de instancias que satisfacen las condiciones de la consulta expresada por el usuario (ver Figura 7). Aniceto incluye un módulo especializado para presentar esta información y permitir que el usuario recorra los resultados, visualice las unidades de información, y navegue entre unidades. Este módulo está basado en el trabajo de investigación realizado en la herramienta Pegasus [2].

El módulo de visualización muestra instancias de la ontología

RESULTADO DE LA BÚSQUEDA

Página 1 de 4 1 | 2 | 3 | 4

[Noticia]
 20-Nov-2003 [Javier Marías ganador del Premio Salambó, 2002 con la novela Tu rostro mañana 1. Fiebre y lanza](#)

[Análisis fundamental]

Nombre del valor	Recomendación	Riesgo	Precio	Fecha
Abertis	Neutral	Bajo	13	01/12/2003

[Análisis técnico]

Nombre del valor	Recomendación	1 ^{er} soporte	resistencia	1 ^a Fecha
Abertis (cierra: 11,78)	Mantener	11.24	12.29	09/12/2003

[Indicadores]

Día	Indicador	Periodo	Datos reales	Previsión	Datos anteriores
20-Nov-2003	Balanza comercial	octubre		941 mm	1.102 mm

[Concurso]
 Del: 11-dic-2003 [Elaboracion lista potenciales proveedores para obras y asistencia tecnica en](#)
 Al: 12-dic-2003 [suministro agua y saneamiento de Posadas y Garupa \(extension red suministro: 77 km en Posadas y 143 Km en Garupa ; extension red saneamiento,341 km en Posadas y 60 km Garupa\)](#)
 [Todos los sectores]

[Encuentros empresariales]
 Del: 11-dic-2003 [Conferencia-Foro Empresarial sobre Irak](#)
 Al: 12-dic-2003

[Ferias]
 Del: 11-dic-2003 [Conferencia-Foro Empresarial sobre Irak](#)
 Al: 12-dic-2003

Figura 7. Resultado de la búsqueda en Aniceto.

mediante páginas web generadas dinámicamente. Cada clase de instancia se presenta de forma distinta, mostrando de forma selectiva sus datos y relaciones con otras instancias (ver Figura 8). En lugar de codificar este tratamiento en un programa, Aniceto permite definir independientemente la presentación de cada clase de la ontología mediante uno o varios modelos de visualización por clase.

El modelo de presentación de cada clase establece las partes de una instancia que se han de mostrar, en que orden y con qué aspecto. Este modelo se define con un lenguaje razonablemente sencillo que permite referenciar y recorrer fácilmente las partes de la red semántica que se han de visualizar. El motor de presentación selecciona dinámicamente la vista apropiada para una instancia en el momento de presentarla, según la clase de instancia y, en su caso, otras condiciones. El módulo de visualización se ocupa asimismo de presentar en la misma página las instancias relacionadas con la que se está visualizando, o bien generar hiperenlaces a éstas, para navegar por relaciones de la ontología.

El lenguaje de presentación incluye la posibilidad de expresar condiciones sobre el modelo de usuario, las características del dispositivo de acceso, el estado de la aplicación, o las características de la propia información a presentar, de modo que cualquier aspecto de la presentación se puede adaptar dinámicamente al contexto de ejecución. Estas condiciones se pueden aplicar a la selección de un modelo de presentación u otro para una instancia, o a nivel más detallado, determinar el aspecto

de pequeñas partes de la presentación, la inclusión o no de fragmentos, la generación de hiperenlaces, la selección de unas u otras componentes de página (listas, tablas, árboles, etc.).

Actualmente en Aniceto se han definido tres modelos de presentación por clase: vista extendida, para mostrar una instancia con máximo detalle en una página; vista reducida, para presentar listas de instancias, por ejemplo las que resultan de una búsqueda; y vista mínima, por ejemplo la cadena de texto a utilizar en los enlaces a una instancia.



Figura 8. Instancia de Aniceto. En este caso un Análisis Fundamental

Los modelos de usuario que se han puesto en práctica hasta el momento reflejan a) perfiles profesionales y b) perfiles de suscripción de los consumidores de contenidos generados por la empresa. El perfil de suscripción conlleva permisos de acceso a unas u otras partes de la ontología, como consecuencia de lo cual se ofrecerá u ocultará el acceso a las distintas áreas de información. El perfil profesional conlleva una escala de interés por los distintos temas y tipos de material, lo que determina el orden (prioridad), y la cantidad de información que se muestra al usuario según la tipología y temática relevantes para su perfil.

En cuanto al diseño del interfaz web, hemos tenido que elegir entre diversas alternativas en cuanto a la forma de representar información de la propia ontología (no de sus instancias), planteándonos incluso la necesidad de crear una ontología de visualización para tratar las diversas posibilidades. Por ejemplo, en el caso de las clases derivadas de *CategoríaAniceto*, mostradas en la parte izquierda de la Figura 9, ¿cómo se muestran las clases?. Nos planteamos las siguientes alternativas:

- Orden alfabético. Poco útil para el usuario. Es la forma habitual en los editores de ontologías, sin embargo, puede que los usuarios también asuman este orden.
- Proximidad conceptual. Obliga a definir una métrica de distancia en el espacio de los conceptos. Estas medidas de distancia tendrán que darse entre todos los conceptos de la ontología. Probablemente es la más útil pues nos permite mostrar visualmente próximos conceptos que están conceptualmente próximos.

Otro tema interesante es cómo representar una ontología. Si en nuestro caso tuviésemos que mostrar *CategoríaAniceto*, con sus 105 clases derivadas, ¿cómo podríamos hacerlo?

- Con un árbol: es la forma tradicional en las herramientas de diseño de ontologías. Pero la topología de una ontología es un grafo por lo que habrá situaciones en las que una clase tenga que aparecer varias veces en el árbol.
- Con un grafo: esto obliga a tener en las aplicaciones web unos sistemas de generación de gráficos sofisticados. Aunque estos sistemas están bastante desarrollados y extendidos (por ejemplo Flash), obligan a disponer de un equipo de desarrollo menos habitual que si se realizan con elementos tradicionales como Javascript.

En cualquiera de los dos casos, cuando hay muchos elementos como es el caso de *CategoríaAniceto*, consideramos que no se puede aturdir al usuario mostrándole todas las opciones a la vez. Nos hemos planteado como solución darle dos vistas: una amplia, de ubicación completa, en forma de “camino de migas” (Empresas > Análisis de mercados > ... > Mercados emergentes en la figura 9); y otra de corto alcance pero detallada donde se muestran sus clases hermanas, padre(s) e hijas (parte izquierda en figura 9). Incluso con esta solución, puede que tengamos que mostrar demasiadas clases a la vez, por lo que el criterio de distancia entre conceptos tratado anteriormente retoma utilidad, permitiendo políticas de visualización del estilo: “mostrar un máximo de 3 niveles con un máximo de 2 clases hermanas siendo estas las más próximas”. Estos valores pueden ajustarse al nivel de experiencia del usuario, ser modificables por éste y, en cualquier caso, el usuario debería tener forma de ver todo el grafo a la vez.



Figura 9. Búsqueda semántica. En este caso se buscan noticias que estén en la categoría Mercados emergentes.

5. Conclusiones

La primera conclusión a destacar es que el uso de las técnicas de la Web Semántica descritas en este artículo está permitiendo a AFI el desarrollo de nuevos productos para sus clientes así como una mejora importante en el diseño e integración de sus aplicaciones. Destaca la búsqueda semántica, ahora más precisa y expresiva que la anterior y la aparición de nuevas relaciones entre elementos que antes no se encontraban relacionados en el portal web de cada cliente. En la fecha de escritura de este artículo aún

no ha terminado el proyecto por lo que no disponemos de los datos de aceptación por parte de los clientes reales.

La usabilidad del interfaz avanzado de consultas nos preocupa especialmente, pues si no le resulta intuitivo y sencillo al usuario, no lo usará y volverá a la búsqueda simple, desaprovechando sus ventajas.

En el momento de escribir este artículo estamos realizando pruebas de rendimiento de los MDBOs frente a las BDs tradicionales, centrándonos en los detalles técnicos que indicamos a continuación:

- Velocidad de las consultas en RDQL
- Coste del razonador
- Tamaño de la BD usada por el MDBO (ver Tabla1).

Aunque preliminares, nuestra experiencia en el proyecto Aniceto nos permite adelantar los datos que se muestran a continuación.

Tabla 1. Rendimiento en Aniceto

	BD tradicional	BD MDBO	BD MDBO con inferencia
Tamaño	~600MB	~700MB	~1GB

Los datos de la Tabla anterior corresponden a 0,25 millones de instancias (noticias) compuestas de 1,6 millones de sentencias RDF.

El desarrollo de este proyecto contribuye al afianzamiento de la Web Semántica por ser esta una disciplina emergente [5]. Para la continuación de este proyecto planteamos la utilidad de crear Servicios Web Semánticos. Los Servicios Web están sirviendo para la integración de aplicaciones, por lo que su adaptación a la filosofía de la Web Semántica se ha convertido en un tema prioritario de investigación y desarrollo. Con estos servicios web semánticos, los agentes personales se pueden convertir en una realidad a medio plazo.

6. Agradecimientos

El trabajo presentado en este artículo está financiado por el Ministerio de Ciencia y Tecnología a través de los proyectos Aniceto (FIT-150500-2003-309) y Arcadia (TIC2002-1948).

7. Bibliografía

- [1] Alonso. J.L., Carranza C., Castells P., Foncillas B., Lara R. and Rico M. Semantic Web Technologies for Economic and Financial Information Management. Poster 2nd International Semantic Web Conference 2003.
- [2] [Castells, 2001] P. Castells and J. A. Macías. An Adaptive Hypermedia Presentation Modeling System for Custom Knowledge Representations. World Conference on the WWW and Internet (WebNet'2001). Orlando, 2001.
- [3] [Berners-Lee 2001] T. Berners-Lee, J. Hendler, O Lassila. The Semantic Web. Scientific American, May 2001.
- [4] T. R. Gruber. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2), pp. 199-220, 1993.
- [5] S. Haustein and J. Pleumann. Is Participation in the Semantic Web too Difficult? International Semantic Web Conference (ISWC'2002). Sardinia, Italy, 2002.
- [6] Jena y RDQL: <http://www.hpl.hp.com/semweb/>
- [7] D2R: <http://www.sourceforge.org/d2r-map>
- [8] Sesame: <http://www.hpl.hp.com/semweb/>
- [9] Protégé: <http://protege.stanford.edu/>
- [10] SHOE: <http://www.cs.umd.edu/projects/plus/SHOE/>
- [11] RDF: <http://www.w3.org/RDF/>
- [12] RDFS: : <http://www.w3.org/TR/rdf-schema/>
- [13] DAML: <http://www.daml.org/>
- [14] OWL: <http://www.w3.org/2001/sw/WebOnt/>