

POTNIA: UNA HERRAMIENTA PARA DIRECTORIOS TEMÁTICOS BASADA EN DUBLIN CORE Y TOPIC MAPS

Piedad GARRIDO PICAZO,¹ piedad@unizar.es

Jesús TRAMULLAS SAZ,² tramullas@unizar.es

1. Universidad de Zaragoza. Departamento de Informática e Ingeniería de Sistemas (Zaragoza, España).
2. Universidad de Zaragoza. Departamento de Ciencias de la Documentación (Zaragoza, España)

Resumen

Se muestra el desarrollo de una herramienta de tratamiento de información, nominada *Potnia*, enmarcada en el Proyecto de estudios y análisis EA 2003-52, financiado por el Ministerio de Educación, y basada en la integración del estándar propuesto por la DCMI (Dublin Core Metadata Initiative) y el paradigma de los *topic maps* registrado en la norma ISO 13250:2003. La propuesta se divide en seis apartados: una introducción donde se describen los fundamentos y objetivos del proyecto; un segundo apartado donde se muestra la hipótesis gráfica del trabajo; un tercer apartado que comenta brevemente la metodología seguida en el desarrollo de la herramienta; un cuarto apartado que describe la integración de ambos lenguajes de etiquetado sentando las bases de su nexo de unión: el lenguaje de etiquetado RDF (Resource Description Framework); un quinto apartado que describe las mejoras con respecto a una versión previa de la herramienta centrándose en aspectos tales como la visualización y recuperación de la información y mejoras en aspectos de seguridad ante el manejo de la misma por diferentes perfiles de usuario; y una última sección con unas conclusiones en las que se hace una reflexión acerca de la incorporación de estos lenguajes de etiquetado a los sistemas de recuperación de información y su utilización de forma aislada o combinada.

Palabras clave

Potnia, directorios temáticos, *topic maps*, metadatos, RDF, XTM, Dublin Core Metadata Initiative, recuperación de la información, visualización de la información, web semántica

1 INTRODUCCIÓN

En el marco del Proyecto de Estudios y Análisis EA 2003-52 sobre directorios temáticos,³ financiado por el Ministerio de Educación (TRAMULLAS 2003), se detectó la necesidad de crear una herramienta de software libre que facilitase la creación y mantenimiento de este tipo de recurso de información digital. En este contexto, y determinada la necesidad de una herramienta para atender a los futuros desarrollos del tratamiento de información en entornos XML, así como a nuevas técnicas de visualización de grandes volúmenes de información, se diseñó e implementó la herramienta de software libre *Potnia* que se distribuye bajo Mozilla Public License.

Se abordó el diseño e implementación de la primera versión con las siguientes premisas de trabajo:

1. utilización de programación no-declarativa;
2. soporte de un sistema gestor de bases de datos relacional;
3. no utilización de técnicas avanzadas de visualización de la información;
4. estructura de descripción de los recursos de información acorde con la norma ISO 15836 (DCMI 2003).

Finalmente, el producto resultante fue llamado *Potnia*. Puede encontrarse una aplicación de prueba en <http://imhotep.unizar.es/potnia>

Sitios de distribución de *Potnia*

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

<http://freshmeat.net/projects/Potnia>

La contribución de esta propuesta surge a partir del planteamiento del rediseño de *Potnia*. Una vez operativa la primera versión, comprobado su correcto funcionamiento y aceptación en los distintos colectivos que la han utilizado, se plantea el desarrollo de una segunda versión. La evaluación de la primera versión de *Potnia* se ha llevado a cabo a partir de las sugerencias y comentarios remitidos por correo electrónico de distintos colectivos y particulares. El equipo de desarrollo era totalmente consciente de que en la primera versión se habían dejado varios aspectos sin desarrollar por la falta de tiempo para su implementación y debido a su complejidad.

Una vez evaluada la utilidad del proyecto, se plantea el rediseño en base a una serie de mejoras de diferentes aspectos de diseño centrado en el usuario como la recuperación y visualización de la información, y aspectos de seguridad. Para poder cumplir estos objetivos, se estuvieron

3. Koch (2000) ha propuesto la siguiente definición: «Subject gateways are Internet-services which support systematic resource discovery. They provide links to resources. The service is based on resource description. Browsing access to the resources via a subject structure is an important feature.»

barajando distintos lenguajes de etiquetado para representar la información. Entre ellos, se encontraban OWL (*Ontology Web Language*), XML (*eXtensible Markup Language*) y XTM (*XML Topic Maps*), lenguajes de etiquetado especialmente adecuados para el tratamiento de metadatos, y que ofrecen nuevas y potentes opciones frente a las carencias y limitaciones en el acceso textual tradicional a los recursos de información digital. Estos lenguajes pueden ser integrados en distintos tipos de repositorios de información como alternativa al modelo relacional de base de datos orientado a objetos o con formato XML. Con este enfoque se puede dar un tratamiento más perfeccionado a la metainformación almacenada en la base de datos, ya formateada siguiendo la norma ISO 15386, y, por supuesto, mejorar la tarea de visualización de la información. Dicha tarea dejará de ser simplemente gráfica y estática, y utilizará un lenguaje que sea capaz de representar la semántica inherente a las interrelaciones establecidas entre los objetos. Por último, y desde el punto de vista de la seguridad, se mejorará la aplicación incluyendo sesiones. Dicha modificación afecta directamente al diseño y programación de la interfaz gráfica de usuario.

Para ello, se tuvo que replantear el diseño inicial de todo el proyecto (ver figura 1) e incorporar las siguientes tecnologías al desarrollo del proyecto:

- *topic maps*;
- base de datos en XML;
- visualización: mapas conceptuales o *topic maps*.

2 HIPÓTESIS DE TRABAJO

El planteamiento del proyecto queda descrito en la siguiente hipótesis gráfica de trabajo:

La hipótesis de trabajo tiene un núcleo central, la aplicación del paradigma de los *topic maps* (ISO 13250:2003). Este paradigma conlleva no sólo una forma de etiquetar la información que favorezca su posterior tratamiento y recuperación, sino una nueva filosofía de trabajo. Se ha apostado por el uso de este paradigma porque se trata de demostrar que es la herramienta más apropiada para las mejoras que se quieren implementar, ya que ofrece la especificación adecuada para que con el uso de un solo paradigma se puedan resolver todas las carencias anteriormente planteadas. Esta especificación es la XML para *topic maps* (XTM 2001), que corresponde al lenguaje de etiquetado utilizado para que el paradigma de los *topic maps* sea legible por un computador. Tal y como se muestra en la figura, se dispone de una colección de recursos de información, en este caso, la recopilación de información relativa a directorios especializados. Esta información es etiquetada haciendo uso de la notación de XTM. Esta metainformación etiquetada es almacenada en un repositorio de información en base a unos criterios. Los editores de XTM y XML permiten establecer una conexión entre el sistema gestor de base de datos

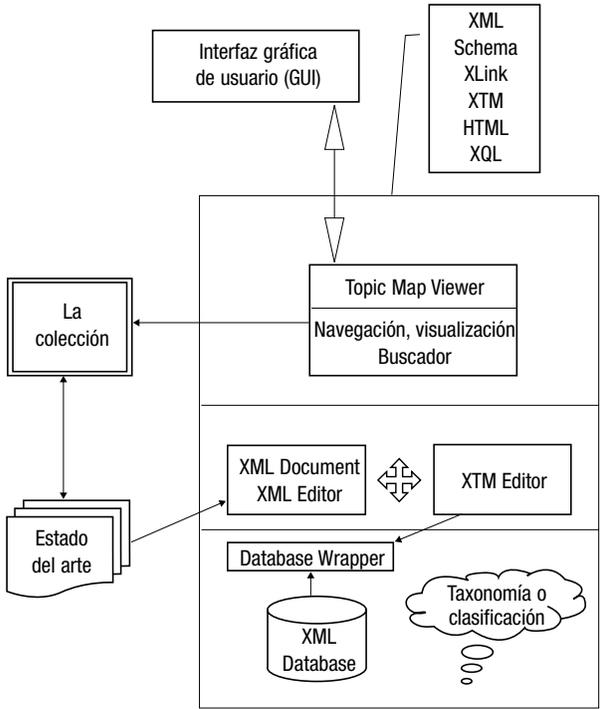


FIGURA 1. Hipótesis de trabajo

utilizado y la interfaz gráfica de usuario, haciendo uso del *database wrapper*, además de facilitar la tarea de etiquetado de la colección. Por último, comentar que la interfaz gráfica de usuario utilizará la información en XTM que acompaña a la colección para la construcción de una interfaz visual (que no textual), en base a unos criterios de navegabilidad y acompañada de un buscador rápido y eficaz, gracias a la incorporación de los *topic maps* al proyecto a lo largo de todo su desarrollo.

3 METODOLOGÍA

La metodología de trabajo seguida para el análisis, diseño e implementación, tanto de la primera como de la segunda versión de *Potnia* sigue las pautas marcadas por la ingeniería de requerimientos. La ingeniería de requerimientos cumple un papel primordial en el proceso de producción de software, sea de la disciplina que sea, ya que trata un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad y sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema. De esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas. Existe una diversidad de estándares y de técnicas al respecto. La utilización de una u otra depende del tamaño del proyecto y del modelo de proceso de software utilizado para el ci-

clo de desarrollo. En el caso que nos ocupa, la metodología seguida ha venido marcada por la guía IEEE Std.830-1984 (ANSI 1984), de la que se extraen cinco actividades principales:

1. Análisis del problema: entender las verdaderas necesidades del rediseño de *Potnia*.
2. Evaluación y negociación: descubrir los problemas potenciales, por ejemplo la elección de un paradigma de programación bastante novedoso y del que no se dispone de toda la información que sería deseable.
3. Especificación de requisitos de software: en la SRS se definen todos los requerimientos de hardware y software, diagramas, modelos de sistemas y cualquier otra información que sirva de soporte y guía para fases posteriores. Gracias a esta etapa del diseño, se puede afirmar que la arquitectura software de dos capas utilizada en la primera versión se puede usar sin ningún problema para el desarrollo de la segunda versión.
4. Validación: revisar el cumplimiento de las características de la especificación de requisitos. Control del cumplimiento de los estándares de calidad.
5. Evolución: un proceso externo que tiene lugar a lo largo del ciclo de vida del proyecto y que pretende comprender mejor el desarrollo de las necesidades de los usuarios. Gracias a esta fase, durante la implementación de la primera versión se detectaron deficiencias, que serán subsanadas en la segunda versión. Todo ello, desde el punto de vista de las necesidades del usuario y complementando las perspectivas de los desarrolladores.

En este momento, el desarrollo de la segunda versión se encuentra en fase de evolución, dado que el software se encuentra en plena fase de programación y ya se dispone de un prototipo.

4 INTEGRACIÓN DUBLIN CORE Y TOPIC MAPS

La integración de Dublin Core en el paradigma de los *topic maps* ha podido llevarse a cabo gracias al modelo de datos formal RDF⁴ (*Resource Description Framework*). El modelo de datos RDF proporciona un marco abstracto y conceptual para definir y utilizar metadatos. Metadatos que puedan ser legibles e interpretados por un computador. De hecho, se diseñan metadatos con el objetivo de ser procesados e interpretados por procesos automatizados tales como los agentes software y los sistemas de búsqueda. El RDF sirve para describir la semántica, pero necesita también una sintaxis concreta para crear e intercambiar metadatos. Esta especificación es conocida como RDF/XML, serialización que utiliza el XML (*eXtensible Markup Language*) codificado como su sintaxis de intercambio.

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

Tal y como se ha indicado, el RDF provee un modelo de metadatos con base sintáctica y semántica, pero no provee un conjunto de propiedades por sí mismo. Es decir, el RDF no define un autor, un título, etc. Ese trabajo queda delegado y por ello entran en juego iniciativas como la DCMI, la construcción de vocabularios RDF, la utilización de ontologías, o el paradigma de *topic maps*.

Los vocabularios definidos por la DCMI usan el RDF en su lenguaje de especificación. Un claro ejemplo de integración de los *topic maps*, Dublin Core y el RDF lo tenemos en el siguiente fragmento de código extraído de Park (PARK 2003).

```
<topic xlink:type=«extended» ... ///... >
  <resource xlink:type=«locator» xlink:href=«... ///...»
    rdf:type=«dublinCore»>
    <dc:creator>Dr Livingstone</dc:creator>
    <dc:language>English</dc:language>
    ... ///...
  </resource>
</topic>
```

Una vez introducido y estudiado en profundidad su nexo de unión, para la segunda versión de *Potnia*, únicamente interesa la integración de Dublin Core y *topic maps*. Tal y como Decambre y Bowers (2000) muestran en un estudio de información sobrepuesta basada en modelos, existen muchas similitudes entre las capas estructurales presentadas por el XML, el RDF y los *topic maps*. De esta forma, es perfectamente posible y correcto, representar la mayoría de las estructuras RDF haciendo uso de sintaxis de los *topic maps* y viceversa. Sin embargo, en este último sentido (representar los *topic maps* con estructuras RDF), se pierde parte de la semántica. Como el objetivo principal de la aplicación es proporcionar un mayor grado de precisión en los resultados de las búsquedas, no interesa la pérdida de semántica. Por este motivo, ha decidido utilizarse la estructura y la sintaxis de los *topic maps*, por tratarse de un paradigma más actual, flexible y abstracto.

En resumen, por un lado se encuentra la representación de la información garantizada con un lenguaje predecible, con etiquetas intuitivas siguiendo el estándar DCMI. Asimismo, se trata de una representación de la información que facilita al máximo el acceso al recurso. Y por otro lado, se encuentran los *topic maps* que, al igual que el RDF, sirven tanto para la descripción como para facilitar la interconexión entre distintos métodos de descripción. En el siguiente apartado, se analizará brevemente el paquete de mejoras pensado para la segunda versión de *Potnia*, que se centrará en aspectos tales como mejoras en la visualización de la información, mejoras en la herramienta de búsqueda e inclusión de aspectos de seguridad en la aplicación.

5 PAQUETE DE MEJORAS

5.1 Visualización de la información

Tal y como Ben Shneiderman comentaba en la International Conference on Information Visualization celebrada en Londres en el año 2001, existe un gran interés por el diseño de presentaciones virtuales de información proveniente de diversas fuentes. El XML sería un ejemplo de lenguaje de etiquetado a utilizar en este caso, debido a la heterogeneidad (formato, origen, etc.) de los datos que suele manipular.

Entendiendo que la visualización de la información está estrechamente relacionada con la GUI (*Graphical User Interface*), hay que dejar preparada la aplicación en la fase de programación para que sea posible generar representaciones gráficas en una, dos o tres dimensiones, haciendo uso de algunas técnicas de visualización presentadas por Park (2003). Para ello, hay que plantearse las diferentes utilidades que pueden tener los *topic maps* de cara al usuario:

- si el usuario quiere obtener una información específica, haciendo una búsqueda exhaustiva de las ocurrencias de una palabra (KSIEZYK 2000) o,
- si el usuario quiere navegar por un sitio web.

La decisión por una u otra utilidad conlleva inclinarse por el diseño de un tipo de interfaz más simple o más complejo, y arrastra consigo escenarios distintos de cara a la recuperación de información. En la primera utilidad descrita, con una interfaz textual sería suficiente, mientras que para la segunda utilidad la representación y la navegabilidad son características esenciales de una buena visualización. En el desarrollo de la primera versión de *Potnia*, se puso especial cuidado en representar internamente la información de una forma cuidada y acorde a un estándar, con el objetivo de facilitar su posterior recuperación. Sin embargo, a la interfaz sólo se le exigió que cumpliera con los estándares más comunes presentes en herramientas de recuperación de información textual presentes en la web.

En la segunda versión, se ha tratado de interconectar los factores de viabilidad con los factores de desarrollo y que la información se estructure de manera que pueda ser utilizada a posteriori para obtener una representación visual más compleja y dinámica, que no se ciña únicamente al aspecto de devolver una información en base a unas consultas específicas expuestas por el usuario. Debe ser posible navegar por el sitio web de una forma cómoda, amena y sencilla. Entre las opciones actuales de visualización se encuentran gráficos (NVISION 1994), árboles hiperbólicos (MUNZNER 1997), mapas (WILLS 1997) y realidad virtual (si se trata de espacios multidimensionales) (LE GRAND 2000).

Esto ha sido posible gracias a la elección del uso del paradigma de los *topic maps*. La especificación XTM ha sido incorporada en la programación de la aplicación, para conseguir que esté totalmente preparada para

que en el momento oportuno pueda ser adaptable a algún algoritmo de representación visual. La ventaja que ofrece la incorporación de los *topic maps* a la arquitectura software de *Potnia* con respecto a la visualización de la información es que permite la navegabilidad e inferencia mediante estructuras semánticas. Este enfoque no sólo mejora la recuperación de información, sino también la gestión del conocimiento y el mantenimiento de este tipo de estructuras. Llegados a este punto, es interesante recordar la independencia de los recursos informativos, ya que nos permite navegar por nodos que no tienen ningún recurso asociado. En este momento, en este paquete de mejora debe finalizarse el módulo de recuperación de la información ya que, como se ha comentado previamente, las características de la GUI están estrechamente ligadas con el tipo de información que se recupere y que se quiera visualizar.

5.2 Recuperación de la información

La información almacenada en un Sistema Gestor de Bases de Datos se representa mediante un lenguaje de definición de datos, por lo que se podría clasificar como información estructurada. La información en un sistema de almacenamiento y recuperación de la información procede de diversas fuentes heterogéneas, por lo que su representación no se ciñe a ningún lenguaje de definición de datos común. Es decir, que puede tratarse, en gran número de ocasiones, de información no estructurada. Cabe plantearse si realmente la información es no estructurada en su totalidad. Esto no es realmente cierto ya que, hablando desde el punto de vista de la tecnología, se necesitan lenguajes de programación legibles por el computador. Por lo tanto, es aconsejable dejar a un lado el término información no-estructurada y hablar de información semi-estructurada. Un claro ejemplo de un lenguaje de etiquetado de este tipo es el XML (*eXtensible Markup Language*) y también, con otro valor añadido, el XTM (*XML for Topic Maps*). De esta situación de partida, se deduce que los mecanismos de interrogación, más comúnmente conocidos como lenguajes de consulta, van a tener orientaciones distintas y van a diferir en algunos aspectos.

En la primera versión de *Potnia*, se optó por el almacenamiento de información en un repositorio relacional y el uso del SQL (*Structured Query Language*) como lenguaje de consulta. Sin embargo, se trata de un lenguaje de consulta con determinados inconvenientes a la hora de buscar información atendiendo a su semántica. En la segunda versión de *Potnia*, y con la incorporación de los *topic maps*, se ha llevado a cabo un estudio de distintos tipos de gestores de bases de datos, como bases de datos nativas en XML, documentales, orientadas a objetos, etc. Finalmente, por cuestiones de viabilidad, se ha optado por tomar una solución híbrida: una base de datos relacional que almacena información en XML etiquetada en Dublin Core. El objetivo buscado es dotar de dinamismo al sistema gestor de base de datos relacional teórico y aprovechar su organización física en memoria para no perder el rendimiento, en cuanto al tiempo de respuesta, que ofrecía la primera versión de la aplicación. Las

ventajas aportadas por los *topic maps* y la DCMI al proceso de recuperación de la información han sido:

- dotar de semántica a elementos que están en la web, al organizarlos y describirlos, sin necesidad de modificarlos mediante otras vías;
- permitir la fusión con otras estructuras de conocimiento, permitiendo una gestión descentralizada, ya que es un aspecto muy útil para una aplicación de este tipo.

Aparte de un cambio en el gestor de base de datos, se han llevado a cabo, de forma paralela, cambios en la programación del buscador interno de la aplicación. Se ha mantenido la estructura de la búsqueda simple y la búsqueda avanzada, pero se han mejorado aspectos tales como utilización de operadores booleanos combinados en el proceso de búsqueda y la incorporación de operadores de truncamiento. Se han analizado algoritmos basados en lógica difusa y algoritmos genéticos (HERRERA-VIEDMA 2001; CRESTANI 2000) para incorporar algunas de sus características a la programación del buscador realizado en su totalidad con el lenguaje de programación PHP (*Hypertext Preprocessor*).

5.3 Seguridad de la aplicación

Actualmente, en los sistemas de redes y autopistas de la información existe un difícil reto: la protección de la información. Los niveles de seguridad a aplicar a un desarrollo software abarcan desde acciones tan sencillas como el control de sesiones y la validación de usuarios hasta transacciones tan complejas que requieren medidas específicas de seguridad que garanticen la confidencialidad, la integridad y la constatación del origen de los datos.

En la primera versión de *Potnia*, se definieron distintos perfiles de usuario, que obviamente accedían a distintas vistas de la base de datos, pero no se tuvieron en cuenta aspectos de seguridad, ya que no eran uno de los requisitos esenciales de la aplicación en su versión 1.0. Después de recibir los comentarios de diversos usuarios y grupos de trabajo que lo han utilizado, se ha detectado una vulnerabilidad en la aplicación relacionada con la falta de sesiones. El uso de sesiones es un método ampliamente extendido en cualquier aplicación de cierta entidad. Esta vulnerabilidad se ha corregido en la versión 2.0. Como parte de la programación de la aplicación en PHP, se ha aprovechado la flexibilidad de este software para subsanar este problema, ya que permite insertar módulos para almacenar/recuperar/reciclar sesiones. En la práctica, la incorporación de sesiones a la versión 2.0 de *Potnia* incluye numerosas ventajas, relacionadas con la carga de datos de sesiones caducadas. Los *topic maps* también han hecho su aportación al uso de sesiones. Al poder definir perfiles de usuarios, mediante el *scope* y el *theme*, se pueden adaptar a distintas comunidades. De esta manera, se pueden compartir recursos informativos y facilitar el almacenamiento de la información relativa a las sesiones de los usuarios en la base de datos.

6 CONCLUSIONES

Cada vez son más notorias las carencias y las limitaciones en el acceso textual tradicional a los recursos de información digital, debido al aumento de la información digital disponible y a la creciente complejidad de la misma. Los sistemas tradicionales no son capaces de representar la semántica inherente a las representaciones, algo necesario de cara a la web semántica y a los nuevos sistemas de gestión de información para las organizaciones. Los *topic maps* se muestran como la aproximación más ajustada a las necesidades actuales de los usuarios de directorios temáticos, ya que ofrecen respuestas a las dos mejoras principales que se plantean en el rediseño de *Potnia*:

- *sistema de recuperación de información*: se puede integrar la semántica que relaciona los datos;
- *visualización de información*: facilidad de desarrollo de interfaces de acceso a la información, basada en la semántica inherente.

Asimismo, se trata de un paradigma no exclusivo, que facilita la configuración de mecanismos de descripción, tratamiento y seguridad inherentes a la aplicación. Al mismo tiempo, presenta una buena escalabilidad y compatibilidad para adaptarse al creciente número de recursos de información. Como conclusión, cabe destacar que se ha apostado por este paradigma. Sin embargo, una buena pregunta a formular, una vez se ha llegado a este punto, es: Si la base de todo lenguaje de etiquetado es el XML, ¿por qué no usar sólo XML? La respuesta es muy sencilla; el XML permite crear vocabularios propios y específicos para cada situación, crear etiquetas nuevas, etc. Sin embargo, falla en el diseño de escalabilidad para metadatos debido a que:

- El orden de aparición de los elementos en un documento XML es significativo y necesario, pero desde el punto de vista de la filosofía del mundo de los metadatos es antinatural.
- No hay ningún mecanismo que asegure que todos los datos necesarios para realizar una búsqueda estén disponibles. Además, una vez la búsqueda ha sido realizada tampoco se garantiza el orden de aparición de acuerdo con la semántica inherente al elemento a representar.
- El mantenimiento del orden correcto de millones de elementos no deja de ser caro y difícil.

De estas debilidades del XML, se deduce que en el caso de detectar en la fase de evolución de la segunda versión deficiencias que afecten al buen funcionamiento de la aplicación (y dada su versatilidad ya que, como se puede observar en la Figura 1, el paradigma de los *topic maps* se encuentra perfectamente integrado en el eXtensible Markup Language (XML) sería posible utilizarla en combinación con otras técnicas de representación del conocimiento basadas en el mismo lenguaje de marca-

do. Dichas técnicas, al no ser tratadas como excluyentes y al surgir para subsanar los problemas anteriormente mencionados, pueden ser perfectamente combinadas para obtener una mejor solución.

BIBLIOGRAFÍA CITADA

- (CRESTANI 2000) CRESTANI, F.; PASSI, G. (eds.) *Soft Computing in Information Retrieval: Techniques and Applications*. Heidelberg: Physica, 2000.
- (DELCAMBRE 2000) DELCAMBRE, L.; BOWERS, S. *A Generic Approach for Representing Model-Based Superimposed Information* [recurso electrónico]. <<http://www.cse.ogi.edu/~shawn/rpe/bowersrpe.pdf>>. [Consulta 12 junio 2004].
- (GEROIMENKO 2003) GEROIMENKO, V.; CHEN, C. *Visualizing the semantic web: XML based internet and information visualization*. London: Springer, 2003.
- (HERRERA-VIEDMA 2001) HERRERA-VIEDMA, E. «Modeling the retrieval process for an information retrieval system using an ordinal fuzzy linguistic approach». *Journal of the American Society for Information Science and Technology*. 52, 6 (2001), p. 460–475.
- (ISO 13250 2002) ISO/IEC 13250:2002 *Topic Maps* [recurso electrónico]: *Information Technology-Document Description and Processing Languages* <http://www.y12.doe.gov/sgml/sc34/document/0322_files/iso13250-2nd-ed-v2.pdf>. [Consulta 11 septiembre 2004].
- (KSIEZYK 2000) KSIEZYK, R. «Answer Is Just a Question [of Matching Topic Maps]» [recurso electrónico]. 2000. <<http://www.gca.org/papers/xmlurope2000/papers/s22-03.html>>. Comunicación presentada al *XML Europe 2000*, París (junio 2000). [Consulta 12 abril 2002].
- (LE GRAND 2000) LE GRAND, B.; SOTO, M. «Information Management—Topic Maps Visualization» [recurso electrónico]. <<http://www.gca.org/papers/xmlurope2000/papers/s29-03.html>>. Comunicación presentada al *XML Europe 2000*, París (junio 2000). [Consulta 22 octubre 2004].
- (MUNZNER 1997) MUNZNER, T. 1997. H3: «Laying Out Large Directed Graphs in 3D Hyperbolic Space» [recurso electrónico]. <<http://graphics.stanford.edu/papers/h3/>>. Comunicación presentada al *IEEE Symposium on Information Visualization*, Phoenix, AZ (octubre 1997). [Consulta 8 mayo 2003].
- (NVISION 1994) Nvision Software Systems, Inc. *NV3D Technical Capabilities Overview* [recurso electrónico]. <<http://www.nv3d.com/html/tco.pdf>>. [Consulta abril 2003].
- (PARK 2003) PARK, J.; HUNTING, S. *XML Topic Maps: Creating and Using Topic Maps for the Web*. Addison-Wesley, 2003.
- (STUART 1999) STUART K. C., MACKINLAY, J.D. y SHNEIDERMAN, B. *Readings in Information Visualization – Using Vision to Think*. San Francisco: Morgan Kaufmann, 1999.
- (TRAMULLAS 2003) TRAMULLAS, J. *Planificación y evaluación de directorios científicos especializados para Internet: su aplicación como instrumentos de docencia en sistemas de enseñanza y aprendizaje virtual* [recurso electrónico]. <<http://imhotep.unizar.es/invest/directorios>> Proyecto EA-2003-52. [Consulta 12 noviembre 2004].
- (WILLS 1997) WILLS, G. J. «NicheWorks—Interactive Visualization of Very Large Gras» [recurso electrónico]. <<http://www.dia.uniroma3.it/calendar/gd97/>>. Comunicación presentada al *GraphDrawing 97*, Roma (septiembre 1997). [Consulta junio 2003].