

Newtenberg: Un Modelo e Implementación de un Sistema de Publicaciones Digitales en la Web.

Carlos Castillo Ocaranza*

Resumen

Se presenta un marco conceptual y un modelo para el desarrollo de sistemas de información en el Web, que constituye una síntesis de las lecciones aprendidas en el desarrollo de sitios de segunda generación (aquellos que mantienen separados contenido y apariencia) y al mismo tiempo permite una transición continua hacia sitios de tercera generación (pertenecientes a la *Web Semántica*). El modelo contempla documentos genéricos, un multiverso de vistas sobre los documentos y un lenguaje de patrones hipertexto que permite implementar estos conceptos en un sitio Web.

1 Introducción

Los sitios de primera generación son aquellos programados completamente en HTML o con herramientas que permiten la conversión de documentos de otros formatos a esta plataforma hipertextual. Si bien se pueden desarrollar muchos sitios pequeños de esta forma, al crecer la cantidad de información y el número de páginas, la mantención de los mismos se hace insostenible.

Los sitios de segunda generación separan los elementos correspondientes al *contenido* del sitio de los elementos que se relacionan con la *apariencia* del sitio. Típicamente se implementan por medio de plantillas escritas en HTML o XSL [17] que procesan documentos en algún formato general (en la actualidad, XML) y los convierten en una multitud de páginas HTML.

Los sitios de tercera generación incorporan elementos semánticos, para poder ser leídos “tanto por máquinas como por seres humanos” [2]; estos elementos semánticos responden a alguna ontología conocida y por lo tanto es posible construir programas que extraigan de ellos la información específica que requiera cada usuario.

Nuestra motivación principal era crear un sistema que permitiera *administrar contenido* y *propiciar la comunicación*, entendiendo esta última como una coordinación de acciones entre todos quienes interactúan con los sitios Web creados usando este administrador de contenido.

Newtenberg, palabra derivada de la contracción de *New* y *Gutenberg*, expresa cierta continuidad con el trabajo de Johannes Gutenberg, en el sentido de preguntarnos cuáles son los “tipos móviles” en este milenio, que permitirán aprovechar y difundir de mejor manera el conocimiento.

*Departamento de Ciencias de la Computación Universidad de Chile; Newtenberg Publicaciones Digitales Ltd. Correo electrónico: ccastill@dcc.uchile.cl

Partiendo de esa motivación construimos un modelo para el desarrollo de publicaciones electrónicas en la Web¹. El modelo a describir es elegante, simple y -como veremos a continuación- da buenos resultados en la práctica. Las ideas clave son:

- Todo documento lleva en sí toda la información necesaria para ser desplegado en distintos contextos. A un documento con estas características lo llamamos **Éidox**.
- No existe una perspectiva preferente para mirar un documento y todas las miradas sobre él coexisten a la vez. A una navegación con estas características la llamamos **Multiverso**.
- Una página puede definirse como un conjunto de filtros sobre ciertos documentos y que los presentan de cierta manera. A una página estructurada de esta manera la llamamos un Molde con **Cajas**.

El resto de este artículo está estructurado de la siguiente manera: la sección 2 menciona algunos trabajos representativos existentes a la fecha en este tema, la sección 3 describe los **Éidox**, la sección 4 el **Multiverso** y la sección 5 las **Cajas**. En la sección 6 se presenta una implementación de este modelo y un sitio construido con esa implementación. En la sección 7 exponemos nuestras conclusiones y discutimos extensiones a lo planteado.

2 Trabajo Previo

Respecto a enriquecer los documentos con información para diversos contextos, la idea de que el próximo paso es el construir una Web Semántica [2] de relaciones ha ido ganando cada vez mayor aceptación, y ha empujado la aparición de (meta)lenguajes para expresar estas relaciones como RDF [14], SHOE [11] y Ontobroker [6]. Un trabajo interesante acerca de cómo puede usarse esta Web semántica para construir documentos hipermediales puede encontrarse en [3].

Respecto a permitir múltiples miradas, nuestro trabajo se relaciona con el de “Placeless Documents” [7, 8], en el cual el escritorio del usuario, con sus archivos y carpetas, es reemplazado por una vista en colecciones inducidas por propiedades de los documentos. El concepto de Multiverso [12] que utilizamos para la navegación tiene una raíz más profunda que la idea de *múltiples vistas* o *visualización multiforme* [13] usada comúnmente en el área de visualización²

Respecto a cómo automatizar la creación de páginas, una serie de trabajos del equipo de Gustavo Rossi (especialmente [15]), muestran una metodología llamada *Object Oriented Hypertext Design Model* (OOHDM), que contempla tres etapas: modelo conceptual, modelo de navegación y modelo de interfaces. OOHDM está basado en el diseño orientado a objetos, que se realiza en la etapa conceptual. Luego, las relaciones entre objetos son explotadas para crear enlaces en el modelo navegacional.

También WebML [4] describe una aproximación al proceso de creación de sitios Web, que enfatiza los aspectos relacionados con definir formalmente cada página y cada zona dentro de la página. WebML define varios tipos de zona como: unidad de dato, índice, filtro, desplazamiento, etc.

¹Llamados a veces *sitios Web intensivos en contenido*.

²La idea de múltiples vistas surge inicialmente para ser aplicada a la interfaz de un usuario y es extendida más tarde a aplicaciones colaborativas [16] en las que interactúan varios usuarios simultáneamente.

Creemos que nuestro marco es más general ya que armamos una página con un conjunto homogéneo de filtros, cuya funcionalidad no es intrínseca sino que corresponde a la distinción que hace un observador al mirar la página en su conjunto y observar las relaciones entre sus partes. Además en nuestro caso el modelo navegacional surge naturalmente del modelo conceptual y éste es más cercano al de objetos basados en atributos [1] que a un modelo de objetos clásico.

3 Éidox

Un **Éidox** (del griego *éidos*, que significa forma y “x” por XML), es una unidad de información estructurada y autocontenida.

Por *unidad* entendemos un objeto que tiene una alta probabilidad de ser recuperado íntegramente en respuesta a un requerimiento del usuario. Ejemplos posibles de unidad son: un artículo periodístico en el caso de un diario, un producto en el caso de una tienda, un libro en una biblioteca o un disco en un catálogo de música.

La *información* puede ser de cualquier naturaleza (texto, sonido, imágenes, multimedios). Decimos que la información está *estructurada* cuando podemos distinguir sus partes y subpartes en una estructura jerárquica. Esta característica es capturada y reflejada en el hecho de que el éidox está expresado en XML.

Por *autocontenida* entendemos que el éidox contiene toda la información necesaria para ser desplegado en cualquier contexto posible, expresada como un conjunto de propiedades³. El éidox contiene también su epigénesis, es decir, su historial generativo y la discusión generada en torno a ese historial.

4 Multiverso

Un **Multiverso** -en oposición a Universo- es un marco conceptual en el cual se permite y fomenta la posibilidad de mirar un mismo objeto desde distintas perspectivas, haciéndose cargo de los intereses del observador y admitiendo que no existe una única forma o mirada sobre el conjunto de documentos. Rechazar la idea de que la realidad es un dominio definido con existencia independiente, implica que hay tantos dominios de existencia como tipos de distinción el observador pueda realizar (Maturana [12]).

En nuestro modelo, un Éidox participa de este multiverso mediante sus propiedades. Una mirada en el multiverso es una vista sobre una o varias de estas propiedades. La verdad es que muy pocos sitios están desarrollados siguiendo esta línea de razonamiento: una amplia mayoría está estructurado con una sola línea jerárquica, y por lo tanto da soporte sólo a un tipo de distinciones.

El ejemplo típico son catálogos de producto en línea en los cuales un producto, por ejemplo, una bicicleta, aparece bajo la categoría “rodados” dentro de “juguetes”, pero no en la categoría “deportes”, o viceversa; o un sistema legal en que aparece un decreto sobre contaminación de las aguas por petróleo bajo la categoría “agua”, pero no bajo “petróleo”, o un diario en que una noticia

³En lo que sigue, a las propiedades les llamamos también clasificandos, término que pareciera tener más sentido en el contexto de administración de documentos.

sobre la creación de una gran industria en una región determinada aparece en la sección de “noticias regionales” y no en la de “noticias económicas”, y así sucesivamente.

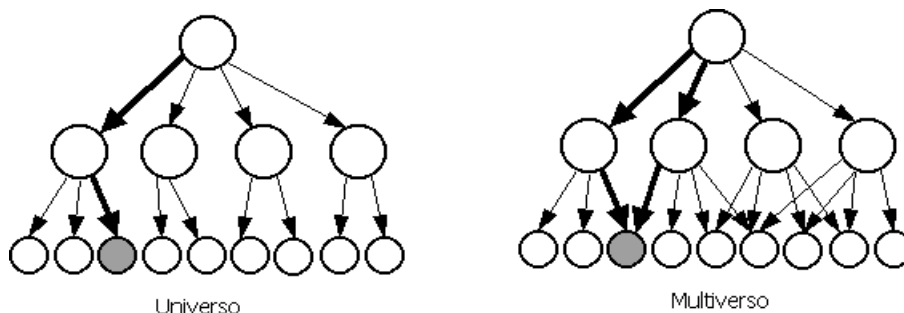


Figura 1: Una navegación basada en multiverso permite en la práctica el acceso a una misma página por distintos caminos.

Este problema es recurrente para los diseñadores de sitios Web. La respuesta en nuestra opinión no es buscar una clasificación perfecta (porque los resultados nunca serán plenamente satisfactorios para todos los observadores), sino modelar desde el comienzo para dar soporte a múltiples clasificaciones; de esta forma se garantiza que el usuario del sitio pueda encontrar la información que necesita de acuerdo a sus propias distinciones, como se muestra en la Figura 1.

5 Patrones hipermedia

La idea de que pueden crearse lenguajes de patrones ⁴ para describir la forma en que se estructura un documento hipermedial ha cobrado bastante fuerza, incluso han aparecido algunos catálogos entre los cuales [10] es uno de los más completos.

5.1 Plantillas: Moldes y Cajas

Un **molde** es una plantilla para un contexto determinado. Este contexto es típicamente una página Web, pero puede ser también un mensaje de correo electrónico, una tarjeta WML o incluso material impreso. Una publicación cuenta con varios moldes que sirven de puente entre los éidox y el mundo exterior.

Los moldes poseen cajas. Una **caja** es un rectángulo en la página que está definido por un conjunto de reglas que definen cuáles éidox seleccionar, cómo representarlos, y con que apariencia gráfica mostrar el resultado.

5.2 Reglas: Selección, Representación y Estilo

Nuestro modelo de generación de cada caja está inspirado en el modelo Cocoon [5] (del proyecto de integración XML/Apache) de generación de documentos. Cocoon establece que existe una cadena de transformaciones desde los datos hasta el usuario, que consta de las siguientes etapas: generación

⁴Como los descritos en [9], relativos al diseño orientado a objetos.

(extracción de los datos que serán usados), proceso (convertir, agregar, consolidar información) y formateo (mostrar los resultados al usuario).

En esta misma línea, una caja dentro de un molde está definida por tres reglas:

- **Regla de Selección:** Indica cuales éidox serán utilizados por la caja y corresponde a una o varias condiciones sobre las propiedades. Ej.: todos los artículos que sean del tema “deportes”, todos los artículos escritos por “Pérez”, todos los productos de menos de un cierto precio, etc.
- **Regla de Representación:** Indica cómo serán desplegados los éidox y corresponde a una o varias propiedades o elementos estructurales. Ej.: mostrar el nombre del producto y su precio, mostrar el titular y la bajada de un artículo periodístico, mostrar un listado con el título y la duración de las canciones de un disco.
- **Regla de Estilo:** Indica qué colores y tipografías deben usarse y corresponde a un fragmento de hoja de estilo. Ej.: títulos en negrita, párrafos en cursiva, etc.

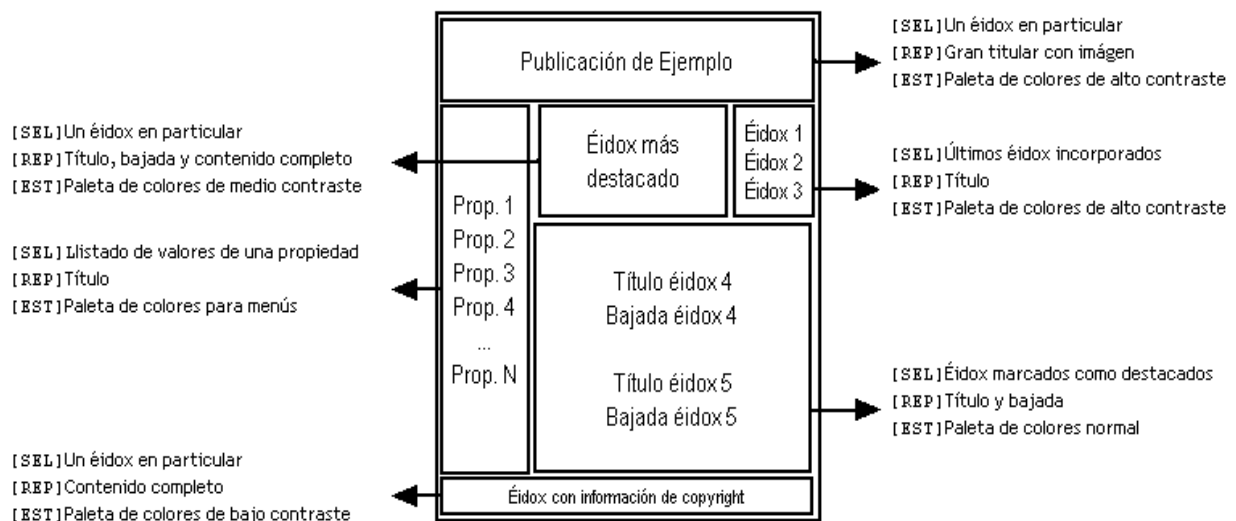


Figura 2: Una molde contiene varias cajas, definidas por reglas de selección (SEL), representación (REP) y estilo (EST).

En la Figura 2 se muestra un ejemplo de molde que puede servir para la portada de una publicación, junto a las reglas que definen cada caja dentro de dicho molde.

5.3 Páginas: Portal, Portada, Portadilla, Artículo

Una **Página** es el resultado de instanciar un molde con un conjunto específico de parámetros. Una página puede provenir de cualquier molde, y no posee a priori ninguna jerarquía ni navegabilidad intrínseca dentro del sitio, pues la navegación se construye combinando cajas que conectan una página con otra.

Es común (aunque no requerido por el modelo) que una página tenga una caja principal o rasgo dominante, más prominente o de mayor tamaño que las demás, que un observador puede distinguir como el área de trabajo de esa página (en contraste con menús, encabezados o ayudas navegacionales que aparecen como accesorias). Tomando en consideración la información presentada en el área de trabajo, distinguimos cuatro tipos de página principales⁵:

- Portal: Es la puerta de entrada a una publicación. Es usualmente una combinación de cajas que muestran artículos y propiedades.
- Portada: Despliega en su área de trabajo un índice de posibles valores para algún clasificando. Ej.: Todas las marcas de producto disponibles, todas las secciones de un diario, todos los departamentos de una tienda, etc.
- Portadilla: Despliega en su área de trabajo un índice de artículos que calzan con alguna o varias propiedades. Ej.: Todos los productos de una determinada marca, lista de artículos de una sección, etc.
- Artículo: Despliega en su área de trabajo un éidox específico.

Esta denominación de páginas es un elemento opcional en el modelo, pues este último es capaz de expresar cualquier patrón de navegación mediante una combinación apropiada de cajas que enlacen las diferentes páginas.

6 Implementación

Newtonberg Engine es un software administrador de contenidos que implementa el modelo que hemos descrito hasta aquí. Permite crear el conjunto de moldes y cajas que define una publicación, administrar éidox que son redactados, clasificados y editados en un entorno colaborativo, y finalmente generar y publicar un sitio Web completo. En la Figura 3 se muestran algunas pantallas representativas de lo que se puede hacer con éste software.

Nuestra implementación consiste en alrededor de 15 clases para administrar los éidox y sus propiedades, implementadas en poco más de 15.000 líneas de código Perl. Los objetos son guardados en una base de datos relacional por medio de una interfaz objeto-relacional desarrollada a este efecto. Los moldes y cajas son construídos en XSL. Existe un generador de éidox que los reconstruye tomando información proveniente de distintas fuentes para producir una hoja de datos XML adecuada para cada página. La arquitectura se ilustra en la Figura 4.

El proceso de generación vía XSLt es más lento que usar plantillas en Perl o PHP, pero facilita el hacer pre-generación de documentos HTML de forma muy ordenada. Se utiliza generación en línea sólo para previsualizar las páginas, las que son luego compiladas y puestas al público en formato HTML. Esto consigue tiempos de acceso para el usuario final mucho mejores que los de cualquier mecanismo que realice el procesamiento de la página en el mismo momento de desplegarla.

⁵Los nombres están tomados de la notación usada comúnmente en diarios o revistas; el proceso de creación de moldes y cajas se parece mucho al que realiza un diagramador al construir una página para un medio informativo.



Figura 3: Algunas vistas sobre la interfaz del Engine.

Ejemplo: SantiagoLimpio.cl

A manera de ejemplo, presentamos un sitio construido con Newtonberg Engine para la Comisión Nacional del Medio Ambiente (CONAMA), Región Metropolitana. El sitio se encuentra en <http://www.santiagolimpio.cl/> y está dirigido a la comunidad interesada en temas medioambientales⁶.

En la Figura 5 se muestra la portada del sitio, que contiene varias cajas mostrando algunos ídiox, algunos nombres de clasificando y algunos valores de un clasificando. Notemos que el sistema es capaz de administrar información en forma de imágenes, de la misma forma en que se administra cualquier documento, y que pueden escogerse discrecionalmente algunos documentos y clasificandos para que aparezcan en una determinada página.

La barra de navegación superior en este caso son nombres de clasificando, que llevan a sus respectivas portadas o índices. En el sitio los documentos son clasificados según cuatro criterios: recurso natural (qué se contamina), fuente de emisión (quiénes lo contaminan), problemática (cómo nos afecta), tipo de documento (si es normativo o de otro tipo), que representan distintas miradas sobre los documentos.

En la Figura 6 se muestra la portadilla del contaminante “Plomo”, a la que se puede acceder, por ejemplo, desde la portada según problemática. Ésta muestra un índice de todos los documentos que tienen “Plomo” como uno de sus valores de propiedad, además de un documento (en el sector

⁶El público ojetivo es muy variado y va desde el ciudadano común hasta empresas de ingeniería que realizan estudios de impacto ambiental, pasando por grupos ecologistas y varias agencias gubernamentales.



Figura 4: Arquitectura del Engine.

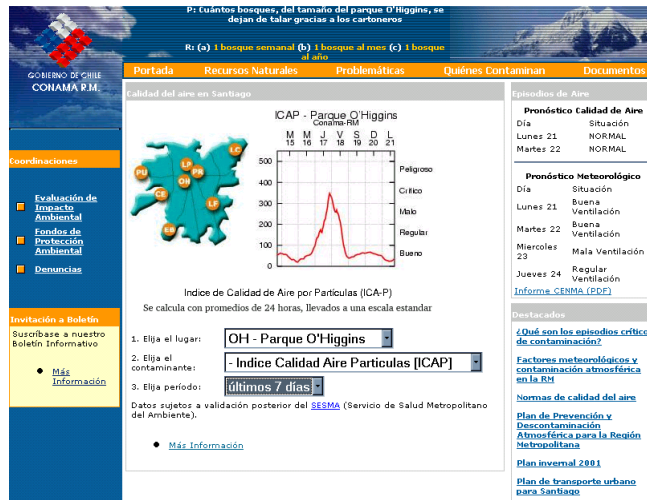


Figura 5: La página inicial de SantiagoLimpio.cl.

derecho) que es la información de contexto esencial acerca de éste valor de clasificando⁷.

En este caso particular hay elementos que se conservan entre página y página, pero en rigor no existe ningún impedimento para que sean todas completamente distintas.

7 Conclusiones

El modelo presentado logra equilibrar simplicidad conceptual, naturalidad al momento de modelar un problema y compatibilidad con los requerimientos tecnológicos de la implementación. Sus características más destacadas son:

- Da soporte a distintas miradas sin obligar a optar por ninguna en especial, lo que permite a cada usuario navegar de acuerdo a sus propias distinciones.

⁷A este documento le llamamos también “ficha maestra” y tiene ciertos clasificandos que la distinguen como tal.

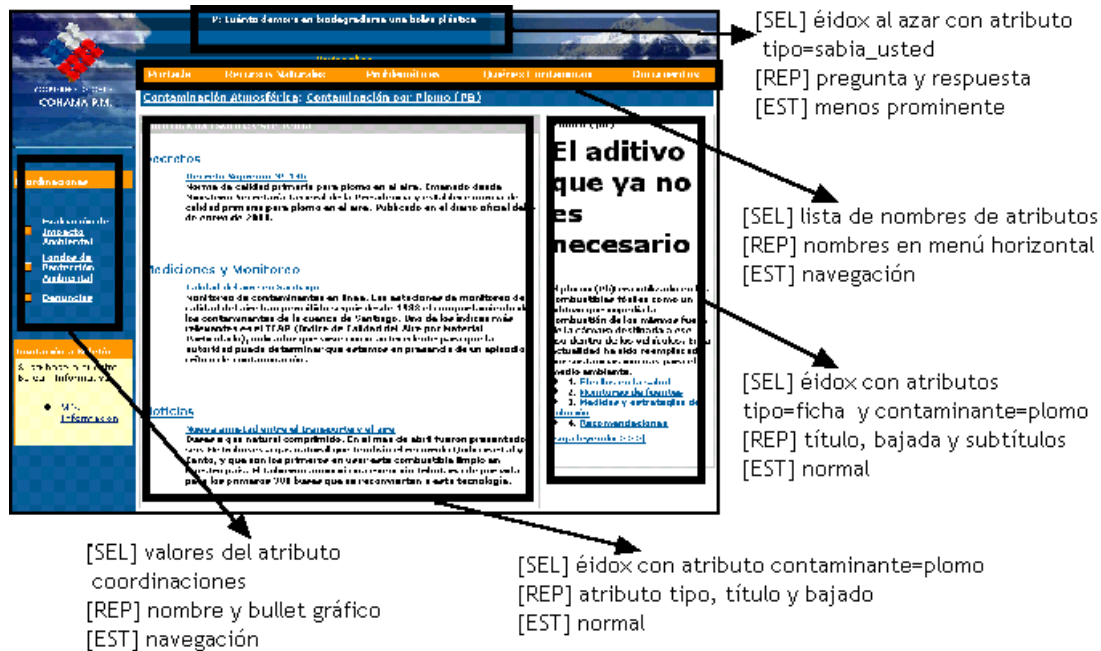


Figura 6: Portadilla para el contaminante plomo.

- Entrega las herramientas para traspasar los requerimientos de un usuario a una implementación concreta rápidamente: basta con determinar las unidades de información y las clasificaciones posibles para poder construir los moldes y generar una primera versión.
- Permite a incorporar nuevo material a un sitio muy fácilmente: basta ingresar el documento y clasificarlo para que automáticamente se generen los enlaces requeridos.

El desafío más importante es el que aparece al considerar documentos heterogéneos; en este caso, hay que contemplar que los documentos pueden tener distinta estructura, pero además, que pueden tener relaciones entre sí con significado, por ejemplo: documentos de tipo “disco” que están relacionados con documentos de tipo “canción” y de tipo “artista”⁸.

Para poder administrar documentos heterogéneos, hemos incorporado desde un comienzo soporte para distintas estructuras en los documentos, asociadas a distintos esquemas XML. También planeamos incorporar relaciones con tipos arbitrarios entre documentos.

Agradecimientos

A todo el equipo de ingenieros y periodistas de Newtonberg Publicaciones Digitales Ltd.; a José Flores de C.E.O. (*Clima de Emprendimiento Organizado*), a Jorge Walters, y a muchas otras personas que han aportado ideas y críticas que nos han ayudado a desarrollar y perfeccionar este modelo.

⁸Éste asunto es el responsable de la complejidad de los modelos presentados en trabajos anteriores, por lo que no es un tema menor.

Referencias

- [1] R. BAEZA-YATES, T. JONES, G. RAWLINS, *A new data model: Persistent attribute-centric objects*, Reporte técnico, 1999.
- [2] TIM BERNERS-LEE, *The Semantic Web*, Scientific American, 2001.
- [3] L. CARR, W. HALL, S. BECHHOFFER, C. GOBLE, *Conceptual Linking: Ontology-based open hypermedia*, Proc. of WWW10, 2001.
- [4] S. CERI, P. FRATERNALI, A. BONGIO *Web Modeling Language (WebML): a Modeling Language for Designing Web Sites*, Proc of WWW9, 2000.
- [5] *Cocoon (in: XML/Apache integration project)*, <http://xml.apache.org>, 2001.
- [6] S. DECKER, M. ERDMANN, D. FENSEL, T. STUDER, *Ontobroker in a nutshell*, Research and Advanced Technologies for Digital Libraries, 1998.
- [7] P. DOURISH, W. EDWARDS, A. LAMARCA, M. SALISBURY, *Using properties for Uniform Interaction in the Presto Document system*, Proc. of UIST 1999.
- [8] P. DOURISH, W. EDWARDS, A. LAMARCA, M. SALISBURY, *Presto: An Experimental Architecture for Fluid Interactive Document spaces*, Xerox Parc, 1999.
- [9] E. GAMMA, R. HELM, R. JOHNSON, J. VLISSIDES *Design Patterns - Elements of Reusable Object-Oriented Software*, Professional Computer Series. Addison-Wesley, 1995.
- [10] D.M. GERMÁN, D.D. COWAN, *Towards a unified catalog of hypermedia design patterns*, 2000
- [11] J. HEFLIN, J. HENDLER, S. LUKE, *SHOE: A Knowledge Representation Language for Internet Applications*, Reporte técnico, University of Maryland, 1999.
- [12] HUMBERTO MATURANA, *Ontology of Observing*, Conference Workbook: Texts in Cybernetics, American Society for Cybernetics Conference, 1988.
- [13] JONATHAN ROBERTS, *Multiple-View and Multiform Visualization*, Proc. of SPIE, 2000.
- [14] *RDF (in: World Wide Web Consortium)*, <http://www.w3.org/TR/PR-rdf-syntax>, 2001.
- [15] D. SCHWABE, G. ROSSI, *An Object-Oriented approach to Web-Based application design*, Capítulo en 'Theory and Practice of Object Systems', ISSN 1074-3224, 1998.
- [16] J. WOOD, H. WRIGHT, K. BRODLIE, *Collaborative Visualization*, Proc. of IEEE Visualization Conference, 1997.
- [17] *XSL (in: World Wide Web Consortium)*, <http://www.w3.org/Style/XSL>, 2001.