

Diseño y montaje de sitios Web utilizando XML y XSI con la herramienta de modelado y consulta Lore para el manejo de consultas

Daladier Jabba Molinares*, Roberto Becerra Bustillo**,
Rodolfo Antonio Cuentas Ahumada***

Resumen

La Web fue diseñada sobre Internet para transferir información de un lugar a otro. World Wide Web (también conocida como Web o WWW) es una colección de ficheros, que incluyen información en forma de textos, gráficos, sonidos y vídeos, además de vínculos con otros ficheros. Los programas informáticos denominados exploradores –como Navigator, de Netscape, o Internet Explorer, de Microsoft– utilizan el protocolo Http para recuperar esos ficheros. La Web maneja grandes cantidades de información, la cual se encuentra dispersa en toda la red. Podemos considerarla como un grafo dirigido en donde los nodos son las páginas Web que pueden encontrarse en cualquier servidor Web y las aristas son los enlaces (hipervínculos) entre ellas. La información circulante por la red generalmente se encuentra sin una estructura definida o irregular, por lo cual es denominada información semiestructurada. Para la realización de consultas especializadas se cuenta con herramientas como Lore, las cuales facilitan el uso de este tipo de información. Lore permite el manejo de la información semiestructurada y la realización de dichas consultas, con una sintaxis parecida a los lenguajes tradicionales como el Sql. Es de vital importancia conocer y aprender manejar las nuevas tendencias en lo que se refiere a desarrollo de sitios Web dinámicos y manejo de cualquier tipo de información en estructuras irregulares (información semiestructurada).

Palabras clave: Html, Dbms, Dtd (definición de tipos de documento), información semiestructurada, Linux, Lore, Sql, Xml, repositorio de datos, servidor Web.

Abstract

The Web was designed on Internet to transfer information from a place to another. World Wide Web (also well-known as Web or WWW) it is a collection of files that include information in form of texts, graphics, sounds and videotapes, besides links

Fecha de recepción: 5 de marzo de 2004
Fecha de aceptación: 7 de mayo de 2004

* Ingeniero de Sistemas de la Universidad del Norte; Magister en Ciencias Computacionales del convenio ITESM-CUTB. Docente tiempo completo del Departamento de Sistemas de la Universidad del Norte. djabba@uninorte.edu.co

** Ingeniero de Sistemas de la Universidad del Norte, 2002. rbecerra@hotmail.com

*** Ingeniero de Sistemas de la Universidad del Norte, 2002. rcuentas10@hotmail.com

with other files. The programs computer denominated browsers - as Navigator, of Netscape, or Internet Explorer, of Microsoft - use the protocol Http to recover those files. The Web manages big quantities of information, which is dispersed in the whole net. We can consider it as a directed graph where the nodes are the Web that can be in any Web Server, paginate and the edges are the connections (links) among them. The circulating information for the net, is generally without a defined or irregular structure, reason why semistructured data is denominated. For the realization of specialized queries, it is had Tools like Lore, which facilitate the use of this type of information. Lore allows the handling of the semistructured data, and it allows the realization of these queries, with a similar syntax to the Traditional languages as the SQL. It is of vital importance to know and to learn to manage the new tendencies in what refers to development of places dynamic Web and manage of any type of information in irregular structures (semistructured data).

Key words: Html, Dbms, Dtd (Document Type Definition), Semistructured Information, Linux, Lore, Sql, Xml, Data Repository, Web Server.

1. INTRODUCCIÓN

Actualmente existen herramientas para el desarrollo y montaje de sitios Web, algunas de ellas un poco rígidas para el posterior mantenimiento de las páginas generadas por los respectivos sitios y otras que en forma dinámica permiten interactuar con la información que se despliega generando un modo fácil de montaje y posterior mantenimiento de los mismos. Lore pertenece a este último grupo; es una herramienta para el manejo de sistemas de bases de datos (DBMS- Database Management System) que utiliza un lenguaje estándar para la representación e intercambio de información en la Web: XML.

Lore «es un manejador de Bases de datos(DBMS) diseñado especialmente para manipular la información semiestructurada. La implementación de esta herramienta ha creado la necesidad de volver a pensar en todos los aspectos que hay que tener en cuenta para un DBMS, incluyendo el control de almacenamiento y la indexación de la información, el procesamiento de queries y las interfaces de usuarios»¹. Es ligero en el sentido que no proporciona todas las características de los sistemas administradores de bases de datos como la administración de transacciones, control de concurrencia o recuperación. Todos los detalles de las capas de esta arquitectura y el modelo de datos, entre otros conceptos, los procederemos a explicar a continuación.

¹ Lore: A Database Management System for Semistructured Data. Disponible en el world wide web <http://www-db.Stanford.edu/lore>.

2. LORE (*Lightweight Object Repository*)

2.1. Generalidades

La cantidad de fuentes de información que se puede acceder electrónicamente está creciendo de manera rápida. Varias de estas fuentes almacenan y exportan datos estructurados o también semiestructurados (muchos de éstos, no obstante, poseen alguna estructura), por lo que una de las motivaciones para usar DBMS diseñados para datos semiestructurados es la fácil integración de los datos de fuentes de información heterogéneas. Un ejemplo de este tipo de información son los archivos HTML que se manejan en la WWW, los cuales usualmente son irregulares o incompletos a pesar de que normalmente tienen algún tipo de estructura, además los datos integrados desde múltiples y heterogéneas fuentes de información a menudo son semiestructurados. Por semiestructurado se quiere decir que aunque los datos pueden tener alguna estructura, ésta no es rígida, regular o completa como la requerida por los sistemas de manejo de bases de datos tradicionales. El manejo de bases de datos tradicionales requiere estrictamente tablas orientadas a datos y se basan en la idea de que un esquema es definido por adelantado y adoptado por todos los datos manejados por el sistema. Mientras que el manejo de los sistemas de bases de datos orientados a objetos permite una gama más grande de estructuras que los sistemas relacionales, ellos aún requieren que todos los datos conformen una esquema de datos predefinido.

El manejo de los datos semiestructurados requiere aspectos típicos de bases de datos, tales como el lenguaje para formar consultas, control de concurrencia y manejo de almacenamiento secundario. No obstante, debido a que los datos semiestructurados no pueden conformar un marco de bases de datos estándar el intentar usar un DBMS convencional para manipular los datos semiestructurados llegará a ser una tarea difícil o hasta imposible. Por todo esto se creó el proyecto Lore (*For Lightweight Object Repository*) en la Universidad de Stanford, donde la meta de este proyecto es proveer esquemas convenientes y eficientes de consulta de datos semiestructurados. El sistema Lore es considerado muy liviano porque el modelo de objetos que utiliza es simple y el sistema mismo no soporta actualizaciones u otras características de las DBMS consideradas más robustas.

2.2. Definiciones

Lore es un sistema manejador de bases de datos (DBMS) diseñado para manejar específicamente información semiestructurada, el cual fue desarrollado por la

Universidad de Stanford en 1995; fue creado originalmente para un modelo de datos llamado OEM (*Object Exchange Model* o Modelo de Intercambio de Objetos), el cual era similar a XML, pero con la diferencia de que OEM no soportaba definiciones de tipos de documentos (DTD's). La nueva versión, usada en este proyecto, ya trabaja con un modelo de datos bajo XML. Como características principales de Lore podemos mencionar las siguientes:

- El modelo de datos de Lore es un modelo de objetos jerarquizado y sencillo llamado inicialmente OEM, aunque el nuevo modelo trabaja sobre XML.
- El lenguaje Lore (Lorel) es una extensión de OQL, el cual introduce expresiones tipo rutas para que las consultas sobre datos semiestructurados sean más poderosas.
- Utiliza resúmenes estructurales de las bases de datos (*DataGuides*) en lugar de esquemas normalizados, los cuales son mantenidos en forma dinámica.
- Permite el manejo de datos en medios heterogéneos.
- Puede servir como un repositorio para almacenar datos semiestructurados y como una máquina integrada para el manejo de consultas.
- Fácil integración de datos cuyas fuentes de información son heterogéneas.
- Incluye un manejador externo de los datos que le permite obtener información desde sitios externos de manera dinámica y durante la ejecución de las respectivas consultas sin que el usuario sepa la diferencia entre datos externos o locales.

En la metodología de diseño de Lore se requiere fundamentalmente de tres pasos:

- Crear la base de datos
- Cargar la base de datos
- Y por ultimo ejecutar Lore sobre la respectiva base de datos

2.3. *Arquitectura*

La arquitectura de Lore y sus diferentes componentes se despliegan en la figura 1. Se puede notar que el procesador de consultas presenta una interfaz textual para interactuar con el manejador de objetos. Procederemos a hablar

de cada una de las capas que intervienen en la arquitectura Lore y los respectivos componentes que la conforman.

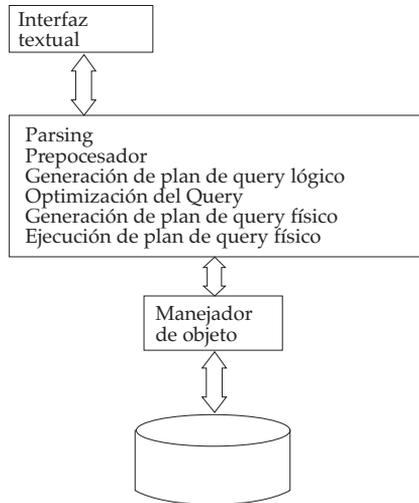


Figura 1. Arquitectura de Lore²

1. **Interfaz textual.** Utilizada por los desarrolladores del sistema para interactuar con la base de datos mediante comandos de consulta.
2. **Capa de compilación de consultas.** Está conformada por el parser, el preprocesador, el generador de plan de consultas y el optimizador de consultas.
 - *Parser.* Es un analizador de sintaxis, el cual acepta la representación textual de una consulta y la transforma en un árbol sintáctico (árbol parser) y la envía al preprocesador.
 - *Preprocesador.* Realiza la transformación de la consulta Lorel a una que tenga la forma de una consulta OQL. A partir de esto se genera un plan de consulta (*query plan*).
 - *Generador del plan de consultas.* Genera conjuntamente con el preprocesador el plan de consultas derivado de su transformación a consulta OQL y es enviado al optimizador.
 - *Optimizador de consultas.* Además de hacer algunas transformaciones en el plan de consultas, decide si el uso de índices es factible o no. El plan de consulta optimizado es entonces enviado a la capa del motor de datos .

² Lore: A Database Management System for Semistructured Data, op. cit.

3. **Capa de motor de datos.** Aquí reside el manejador de objetos OEM, los operadores de consulta, el manejador externo de los datos y varias utilidades.

- *El manejador de objetos OEM.* Funciona como la capa de traducción entre OEM (Modelo de Intercambio de Objetos) y la construcción de archivos a bajo nivel.
- *Los operadores de consulta.* Ejecutan los planes de consulta generados
- *Utilidades:* Manejadores de índices, cargador, manejador de guías de datos.
- *El manejador de datos externos.* Permite la recuperación en forma dinámica de la información desde otras fuentes de datos basadas en consultas efectuadas a Lore; los datos tanto externos como los residentes en Lore son combinados entre sí durante la evaluación de la consulta, siendo esto transparente para el usuario.

4. **Almacenamiento físico.** Lore agrupa los objetos en páginas que conforman el disco físico, cada página tiene un número de slots o ranuras con un único objeto en cada *slot*.

2.4. Manejo de datos

2.4.1. Modelo de datos

El modelo de intercambio de objetos está diseñado para el manejo de datos semiestructurados. Los datos en este modelo inicialmente se diseñaron como un grafo dirigido etiquetado donde los vértices en el grafo eran los objetos y cada objeto tenía un único identificador de objeto (oid), los objetos atómicos no tenían arcos salientes y contenían un valor de uno de los tipos atómicos básicos, tales como *integer*, *real*, *string*, *gif*, etc. Todos los demás objetos podían tener arcos salientes y eran llamados objetos complejos. En el ejemplo desplegado en la figura 2 vemos que un oid es &4, un objeto atómico es &8, cuyo valor es «Smith», además &8, &9, &10 y &11 son objetos dentro de un objeto complejo que es &3. Por último, DBGroup es el nombre que denota al objeto &1, y cualquier objeto que no pueda ser accedido a través de una ruta desde algún objeto se considera que debe ser borrado.

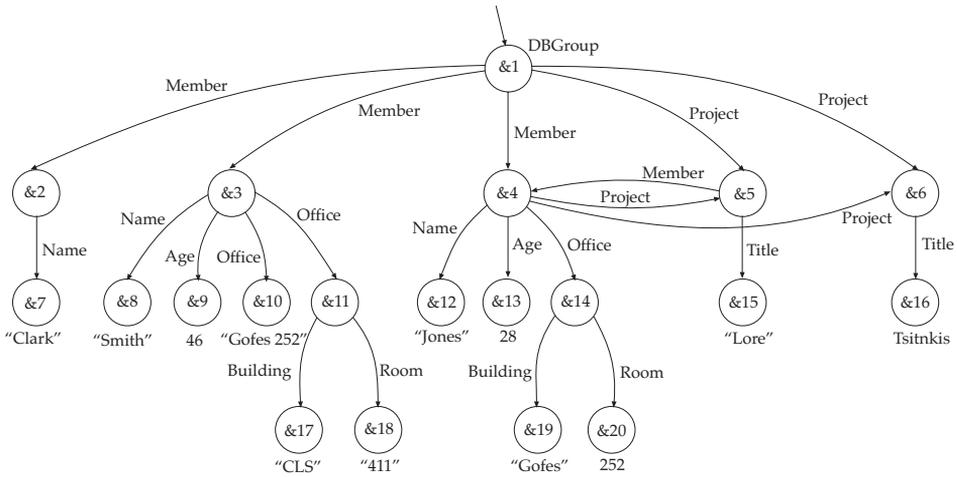


Figura 2. Ejemplo de una base de datos OEM (Modelo de Intercambio de Objetos)³

Este modelo de datos que existía inicialmente se conoce como un modelo de intercambio de objetos (OEM). En una base de datos OEM no existe la representación de un esquema fijo, toda la información esquemática es incluida en las etiquetas, las cuales pueden cambiar dinámicamente, y además su descripción es automática y no hay una regularidad impuesta en los datos; el modelo es diseñado para manejar datos incompletos, así como la estructura y el tipo de heterogeneidad. Para un OEM, dado un objeto X y una etiqueta l, la expresión X.l se refiere al conjunto de los subobjetos l etiquetados de X, donde, si X es un objeto atómico o si l no es una etiqueta saliente de X, entonces X.l es el conjunto vacío. Las expresiones de este tipo con el punto incluido son usadas en los lenguajes de consulta en Lore (Lorel), el cual fue diseñado para trabajar sobre este modelo de datos.

La emergencia reciente en la utilización del lenguaje de marcación extensible (XML) como una nueva norma para la representación de los datos e intercambio de información en la Web ha ocasionado como consecuencia que se le preste una especial atención al modelo de datos (OEM) que inicialmente se utilizaba. Los investigadores han observado una similitud importante entre los modelos de datos semiestructurados y XML. Mientras las similitudes abundan,

³ Lore: A Database Management System for Semistructured Data, op. cit.

algunas diferencias claves exigen cambios al modelo para que los datos semiestructurados puedan soportar XML. El nuevo modelo de datos de Lore basado en XML está conformado por un elemento que es un par (eid,valor), donde eid es un único identificador de elementos y valor es un texto atómico tipo *string* (caracteres) o un valor complejo que contiene cualquiera de los siguientes 4 componentes:

- Una etiqueta tipo *string* que corresponde a la etiqueta XML para ese elemento.
- Una lista ordenada de parejas de nombres de atributos / valores atómicos, donde cada nombre de atributo es un *string* y cada valor atómico tiene tipo atómico, ya sea entero, real, string, etc., o ID, IDREF o IDREFS.
- Una lista ordenada de subelementos entrelazados de la forma (etiqueta, eid), donde etiqueta es un *string*; los subelementos entrelazados aparecen dependiendo de que el atributo sea del tipo IDREF o IDREFS.
- Una lista ordenada de subelementos normales de la forma (etiqueta, eid) donde etiqueta es un *string*; los elementos normales o comunes aparecen debido a los bloques de anidaciones léxicas en un documento XML.

Debemos tener en cuenta que en este nuevo esquema se pueden soportar modos ya sean literales o léxicos. Se hace corresponder un documento XML con el nuevo modelo sin ningún problema, y se ignoran los comentarios y los espacios en blanco entre los elementos etiquetados. Como un caso base, el texto entre etiquetas es traducido a un elemento de texto atómico o también puede ser traducido a un elemento dato complejo tal que:

- La etiqueta del elemento dato es la etiqueta del elemento documento.
- La lista de pares de nombres de atributos / valores atómicos en el elemento dato se deriva directamente de la lista de atributos del elemento documento.
- Para cada valor atributo *i* del tipo IDREF en el elemento documento, o componente *i* de un valor atributo del tipo IDREFS, existe un subelemento entrelazado (etiqueta, eid) en el elemento dato, donde la etiqueta es el correspondiente nombre atributo y el eid identifica el único elemento dato cuyo valor atributo ID está ligado directamente con *i*.
- Los subelementos del elemento documento aparecen en orden como los subelementos normales del elemento dato. La etiqueta para cada subele-

mento dato es la etiqueta de ese subelemento documento, o texto si el subelemento documento es atómico.

Una vez un documento de XML se hace corresponder con el modelo, es conveniente visualizar los datos como un grafo dirigido, ordenado y etiquetado. Los nodos en el grafo representan los elementos datos y los arcos representan la relación elemento y subelemento. Cada nodo que representa un elemento dato complejo contiene una etiqueta y una lista ordenada de parejas de nombre atributo/valor atómico; los nodos de elementos datos atómicos contienen valores *string*. Hay dos tipos de arco diferentes en el grafo: (i) el subelemento normal arco, que contiene la etiqueta del subelemento destino; (ii) arcos entrelazados, etiquetados con los nombres de los atributos de los arcos entrelazados.

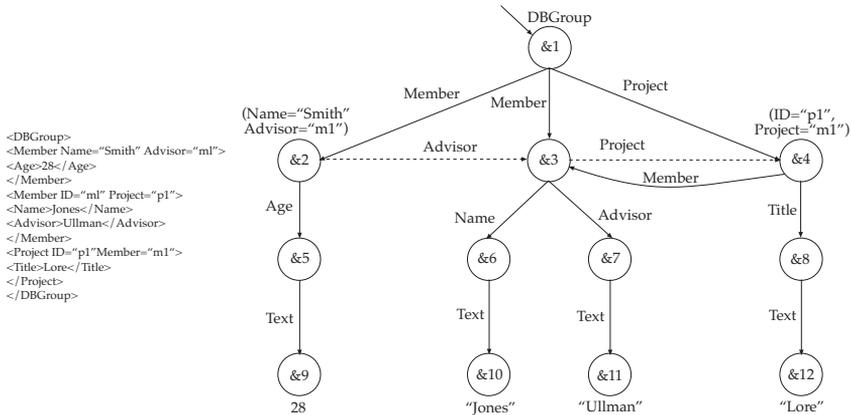


Figura 3. Documento XML y su grafo⁴

Es útil ver los datos XML en uno de dos modos: semántico o literal. El modo semántico se usa cuando el usuario o la aplicación desea ver la base de datos como un grafo interconectado. El grafo que representa el modo semántico omite atributos de tipo IDREF y IDREFS; además, se hace la diferencia entre el subelemento y los arcos entrelazados. El modo literal está disponible cuando el usuario desea ver la base de datos como un documento XML. Los atributos IDREF e IDREFS son visibles como caracteres textuales, mientras los arcos entrelazados son invisibles. En el modo literal, la base de datos es siempre un árbol. La figura 3 despliega un pequeño ejemplo de un documento XML y la representación del grafo en nuestro modelo de los datos. Los identificadores de elementos

⁴ Lore: A Database Management System for Semistructured Data, op. cit.

(eids) aparecen dentro de los nodos y se describen como &1, &2, etc. La pareja nombre atributo/valor atómico es desplegada junto a los nodos asociados (encerrados por {}), con atributos de tipo IDREF en letra cursiva. Los arcos de los subelementos se muestran en líneas continuas y los entrelazados en líneas punteadas, además el orden de los subelementos es de izquierda a derecha. Se debe tener en cuenta que en el modo semántico, la base de datos de la figura 3 no incluye los atributos de IDREF que están en letra cursiva. En modo literal, los arcos entrelazados punteados no son incluidos.

2.4.2. El lenguaje de consulta Lorel

El lenguaje de consulta utilizado inicialmente por Lore era el Lorel, el cual trabajaba sobre el modelo de datos OEM, pero como consecuencia de la migración al modelo de datos XML se le hicieron algunas modificaciones al Lorel para que éste pueda trabajar sobre este tipo de documentos y así explote todos los aspectos de XML, los cuales no se presentaban en OEM. Inicialmente se realizará una descripción de cómo funcionaba el lenguaje de consulta Lorel, para después explicar las respectivas modificaciones que se le hicieron para que funcionara como lo hace actualmente.

Lorel es una extensión de OQL; en la construcción básica de Lorel tenemos las expresiones de rutas simples, las cuales son un nombre seguido por una secuencia de etiquetas, por ejemplo, *DBGroup.Member.Office* es una expresión de ruta simple y su semántica consiste en el conjunto de objetos que pueden ser alcanzados, empezando con el objeto *DBGroup*, seguido de un arco etiquetado *Member* y después otro arco etiquetado *Office*. Las expresiones de ruta también pueden ser usadas directamente en SQL como en el siguiente ejemplo:

QUERY

```
Select DBGroup.Member.Office  
where DBGroup.Member.Age > 30
```

Este ejemplo corresponde a una consulta en SQL que obtiene como resultado las oficinas de los miembros cuya edad es mayor de 30. El resultado de la consulta es la siguiente (aplicado a la figura 8):

```
Office "Gates 252"  
Office  
Building "CIS"  
Room "411"
```

Un principio que se debe tener en cuenta en Lorel es que para escribir una consulta no debe haber preocupación acerca de irregularidades o conocer la estructura precisa de los objetos ni tampoco molestarse por conocer en forma exacta los tipos atómicos (ejemplo, si Age es Integer).

Una vez que la consulta es compilada, el procesador de consultas de Lore rescribe el comando y lo lleva a una forma más elaborada en el estilo OQL; por ejemplo, la consulta escrita con anterioridad queda así:

```

select O
from DBGroup.Member M, M.office O
where exists A in M.Age : A > 30
  
```

El sistema Lore ejecuta el *query* incorporándole algunos aspectos, tales como reglas especiales de acción en grupo; por otra parte, se puede observar que durante el preprocesamiento las expresiones de rutas simples se convierten en consultas que utilizan variables. La versión actual del lenguaje de consulta difiere de la anterior en que para XML las expresiones del tipo rutas sirven también para navegar entre los atributos y los subelementos; además surge un nuevo concepto que es el calificador de expresiones de ruta que se utiliza para distinguir entre los dos (atributos y subelementos) cuando se desea. Se utiliza también el símbolo > antes que una etiqueta para indicar que se aparean sólo subelementos y el símbolo especial @ para indicar que se aparean atributos solamente. La tabla 1 muestra un ejemplo de expresiones de ruta con calificadores aplicados a la base de datos de la figura 3.

Tabla 1
Calificadores de expresiones de ruta

Calificativo	Símbolo	Ejemplo	Equivalencia en modo semántico	Equivalencia en modo literal
Subelementos (Únicamente)	>	DBMember.>Name	&6	&6
		DBMember.>Advisor	&3, &7	&7
Atributos (Únicamente)	@	DBMember.@Name	"Smith"	"Smith"
		DBMember.@Advisor	Empty	"ml"
Ninguno	Ninguno	DBMember.Advisor	&3, &7	&7, "ml"

Los pasos que Lore utiliza para responder a una consulta son:

- La consulta es transformada en un árbol de análisis sintáctico
- El árbol analizador es preprocesado y traducido a una consulta OQL
- Se construye un plan de consulta
- Se realiza una optimización de la consulta
- La consulta optimizada es ejecutada

3. MONTAJE DE UN SITIO WEB CON DATOS SEMIESTRUCTURADOS

3.1. Generalidades

Se puede distinguir varias partes de un escrito cualquiera según el estilo de presentación que éste tenga (un título, el texto del artículo, que a la vez son varios párrafos separados por líneas, titulares, imágenes o simplemente, distancia física en la página).

El hecho de que se pueda distinguir fácilmente estas componentes de un documento, que se llamará estructura del texto, proviene de convenciones tipográficas y de diagramación, a saber, que lo normal es tener el título destacado con un color, un tamaño o una tipografía distinta, separado del texto. Así mismo, es común que la firma tenga algún atributo distinto (por ejemplo, estar en cursiva) y se encuentre al final del documento. A todos estos elementos los llamaremos el formato del texto. La primera observación es, entonces, que el formato permite a los lectores reconocer la estructura del texto. A pesar de que una misma estructura puede dar origen a varios formatos distintos (basta mirar las portadas de los periódicos o revistas para darse cuenta de que existe maneras distintas de representar una estructura básica consistente en un titular y varias noticias con sus correspondientes resúmenes), el problema de recuperar la estructura a partir del formato y poder contestar preguntas como: «¿Hay algún artículo escrito por Pedro Pérez?», «¿Cómo le fue al chino Ríos?» es trivial para cualquier persona. Sin embargo, estas preguntas exceden las capacidades de los sistemas automatizados de recuperación de la información actuales. Todos los que han navegado por Internet y han ocupado un sistema de búsqueda se han dado cuenta de que los resultados de las búsquedas son en varias ocasiones irrelevantes respecto a la pregunta planteada.

Los sistemas de búsqueda obtienen respuestas irrelevantes en gran parte porque no son capaces de reconocer la estructura del documento a partir de distintos formatos. He aquí la importancia de que se pueda reconocer la estructura del documento, y para esto tenemos la herramienta XML, con la cual se

pueden manejar datos semiestructurados. Por otra parte, Lore es una herramienta para el manejo de sistemas de bases de datos (*DBMS-Database Management System*) que utiliza un lenguaje estándar para la representación e intercambio de información en la Web: XML, diseñado especialmente para manipular la información semiestructurada. De esta manera, en el documento XML se pueden tener los datos semiestructurados correspondientes a la información de la base de datos, y de acuerdo con la metodología de diseño de Lore podemos crear y cargar la base de datos a partir del documento XML y ejecutar Lore sobre la respectiva base de datos para realizar las consultas o búsquedas que se necesiten ser visualizadas.

Por último, para la generación de páginas Web que sean entendidas por los humanos y tengan un formato de presentación igual que el HTML, es necesario utilizar una herramienta de XML denominada XSL para poder crear el estilo de la página Web a partir del documento XML sin perder de vista la estructura de los datos. Estas herramientas permitirán mantener el sitio actualizado con la información que se genere dinámicamente en una organización. La importancia de éstas radica en que el usuario que mantiene un sitio no tendrá que preocuparse por los detalles de los contenidos, ni la estructura del sitio, ya que la información se almacenará en repositorios de datos, los cuales deben estar actualizados para que la información que se despliegue también esté al día.

3.2. Estructura de los datos basado en XML

XML es un lenguaje de marcas que ofrece un formato para la descripción de datos semiestructurados; con él podemos definir nuestro propio lenguaje de presentación y, a diferencia del HTML, que se centra en la representación de la información, XML se centra en la información en sí misma. XML no posee etiquetas prefijadas con anterioridad, ya que es el propio diseñador el que las crea a su antojo, dependiendo del contenido del documento. XML describe un conjunto de documentos llamados «documentos XML» y da indicios de cómo debieran ser leídos esos documentos por ciertas aplicaciones. Un documento XML consiste de un conjunto de entidades, que son principalmente texto y marcas. De esta forma podemos definir nuestras propias marcas o etiquetas para el manejo de nuestra información de la base de datos. Supongamos que nos proponemos permitir búsquedas semiestructuradas sobre la portada de un Servicio de Hosting (sitio web ejemplo), abstraigámonos un momento del formato y pensemos en una manera de describir esta estructura. Para conseguirlo, le entregamos al computador el texto de la portada con ciertos marcadores para que él pueda ver la estructura, por ejemplo:

```

<Principal>
<nomprin>Servihosting</nomprin>
<logoprin>C:\logoser.jpg</logoprin>
<dirprin>Barranquilla-Atlantico-Colombia. Calle 93 #51-25</dirprin>
<correoprin>servihosting@hosting.com</correoprin>
<telprin>3458785</telprin>
<bienvenida>ServiHost es una compañía especializada en alojamientos en Internet a
un precio adecuado y con el mejor servicio técnico y atención al cliente que pueda
desear. Es un alojamiento profesional con la tecnología más avanzada para su tienda
virtual o el sitio web de su empresa, no dude que ServiHost es la empresa que necesita.</
bienvenida>
</Principal>
<Empresa>
<nit>10</nit>
<nomemp>Monomeros</nomemp>
<mision>Monomeros es una empresa transnacional dedicada a las exportaciones e
importaciones de diversos productos</mision>
<dir>Via 40 Barranquilla</dir>
</Empresa>
<Empresa>
<nit>11</nit>
<nomemp>Bavaria</nomemp>
<mision>Bavaria es una empresa que se dedica a la fabricacion de bebidas alcoholicas
y otros productos</mision>
<dir>Calle 30</dir>
</Empresa>

```

Un programa computacional puede ver este texto, determinar su estructura, basándose en los marcadores <XXXX> y guardar en su memoria una representación (como un árbol jerárquico, por ejemplo) de dicha estructura, para que después se puedan hacer consultas sobre éste documento utilizando Lore.

3.3. Diseño de estilo y presentación bajo xsl

XML estaría «incompleto» si dejara de lado otro asunto importante: nos sentimos más cómodos cuando el formato no es tan explícito, y nos gusta ver documentos diagramados con criterios estéticos más que informáticos. Desde el punto de vista de quien crea una publicación, sería interesante contar con algún tipo de mecanismo estándar para dar el formato a partir de la estructura.

Para resolver este problema existe el xsl, con el cual podemos diseñar nuestras plantillas de estilo con las cuales pueda procesar la información del documento XML, por ejemplo:

```

<PLANTILLA TIPO=«PORTADA»>
  <formato:CENTRADO>
    <formato:GIGANTE>La Gazetta</formato:GIGANTE>
  </formato:CENTRADO>
  <PROCESAR TIPO=»TITULAR»>
  <PROCESAR TIPO=»NOTICIA»>
</formato:MEDIANO>
  <INSERTAR_FECHA>
</formato:MEDIANO>
<formato:CHICO>
  Chile: $1000, Recargo por flete: $50
</formato:CHICO>
</PLANTILLA>
<PLANTILLA TIPO=»TITULAR»>
  ...
</PLANTILLA>
<PLANTILLA TIPO=»NOTICIA»>
  <formato:GRANDE>
    <TITULAR>
  </formato:GRANDE>
  <formato:CHICO>
    <formato:CURSIVA>
      <RESUMEN>
    </formato:CURSIVA>
  </formato:CHICO>
</PLANTILLA>

```

El marcador `<PLANTILLA> ... </PLANTILLA>` encierra la descripción de lo que se debe hacer con cada elemento estructural; lo mismo para con los marcadores `<PROCESAR>`. Además, hay algunos marcadores que sirven específicamente al propósito de proveer el formato, como `<formato:GRANDE>`, `<formato:CURSIVA>`. Así, la plantilla XSL nos permite darle estilo a nuestras páginas web a partir de documentos XML con datos semiestructurados. Un ejemplo gráfico aplicando estos conceptos se muestra en la figura 4.



Figura 4. Ejemplo desarrollado con XML y XSL

CONCLUSIONES

El sistema Lore, interactuando con XML y XSL, resalta la importancia que tiene la construcción y el mantenimiento de sitios Web con grandes cantidades de datos. Mediante esta investigación hemos mostrado otra alternativa para desarrollar sitios Web en forma dinámica, tal como lo hacen otras herramientas, como es el caso de Araneus y Strudel. Todas estas herramientas trabajan con una metodología que consiste en establecer una separación clara de las actividades inherentes a la construcción de un sitio y hacer posible los cambios en uno de los componentes sin necesidad de afectar a los otros.

Por otra parte, hemos visto que se puede interactuar con Lore teniendo en cuenta que los sitios Web trabajan con información semiestructurada, la cual es manejada por XML, que es el lenguaje estándar para la representación e intercambio de información en la Web, apoyándose en XSL, que es quien se encarga de crear el estilo de las diferentes páginas que conformarán luego el respectivo sitio.

Bibliografía

- DATA ON THE WEB: FROM THE RELATIONS TO SEMISTRUCTURED DATA AND XML. Morgan Kaufmann Publishers, 2000.
- GOMEZ O., Tutorial sobre XML. Citado el 1º de mayo del 2001. Málaga. Disponible en el world wide web <<http://face.el.uma.es/imasd/xml/xml.html>>
- INTERNET Related Technologies. Disponible en el world wide web <<http://tech.irt.org/articles/js072/index.htm>>
- INTRODUCCION XML. Disponible en el world wide web <http://www.tejedoresdelweb.com/avanzado/introduccion_xml/default.f.html>

LORE: *A Database Management System for Semistructured Data*. Disponible en el world wide web<<http://www-db.stanford.edu/lore>>

SEMISTRUCTURED DATA. Disponible en el world wide web<<http://citeseer.nj.nec.com/buneman97semistructured.html>>

XML and XSLT. Disponible en el world wide web<http://www.programacion.com/html/articulo.php?id=tw_xml_y_xslt>

XSL and XML. Disponible en el world wide web<<http://tech.irt.org/articles/js072/index.htm>>

XSL TUTORIAL. Disponible en el world wide web<<http://www.zvon.org/xxl/XSL-Tutorial/Books/>>