Generación de Hiperbases en Internet

José María Martínez Sánchez y Francisco Morán Burgos

e-mail: {jms,fmb}@gti.upm.es
WWW: http://www.gti.ssr.upm.es./~{jms,fmb}/

Grupo de Tratamiento de Imágenes
Depto. de Señales, Sistemas y Radiocomunicaciones
E.T.S.I. de Telecomunicación
Universidad Politécnica de Madrid
Cuidad Universitaria s/n, 28040 MADRID

RESUMEN

Se viene llamando hiperbases a aquellos sistemas que extraen información de una base de datos para generar hipermedia. En el caso de la WWW, las hiperbases existen gracias al interfaz CGI, que permite al cliente invocar a programas del servidor, en lugar de pedirle simplemente ficheros HTML estáticos.

Es evidente la conveniencia de una herramienta para generar automáticamente esos programas que exija al diseñador del sistema un conocimiento mínimo, incluso nulo, del interfaz CGI y del manejo del RDBMS subyacente.

CATALOG, una herramienta de generación de hiperbases desarrollada siguiendo las ideas presentadas en esta ponencia, opera sobre bases de datos relacionales y permite al creador de la hiperbase ofrecer mecanismos de consulta vía WWW sin escribir una sola línea de código y sin modificar en nada la base de datos.

INTRODUCCIÓN

Hasta no hace mucho, multimedia (múltiples medios con relaciones espaciales y temporales que implican sincronización entre los medios) e hipermedia (que añade enlaces lógicos a la multimedia) estaban asociadas a aplicaciones cerradas en soporte CD-ROM, ya que ésta era la única tecnología disponible para combinar acceso aleatorio con alta capacidad de almacenamiento y bajo coste de edición y distribución a gran escala. Si bien la inversión en iniciativas en esta dirección crece, al igual que el número de títulos en CD-ROM, actualmente se está invirtiendo mucho más en iniciativas basadas en redes de telecomunicación.

De entre estas iniciativas, Internet (junto a Infovía en España) es la que más interés suscita, debido a la amplia difusión y aceptación de la *WWW (World Wide Web*, o Tela de Araña Mundial, TAM).

El objetivo principal es proporcionar servicios de acceso a la información en un entorno abierto y heterogéneo. Actualmente, la mayoría de los contenidos accesibles en WWW son conjuntos de páginas estáticas escritas en HTML, relacionadas mediante enlaces estáticos. Esto se puede identificar, al igual que los CD-ROM, con *libros electrónicos*, frente a las *bibliotecas electrónicas*. La desventaja de los libros electrónicos es la necesidad de escribirlos y modificarlos de un modo manual cada vez que se quiere añadir nuevos contenidos. Por su parte, las bibliotecas electrónicas generan dinámicamente las páginas de los diversos datos que contienen, proporcionando además la posibilidad de tener distintas *vistas* de estos datos. El coste de añadir o modificar elementos a una librería electrónica soportada por una base de datos es marginal comparable con el de producción de libros digitales.

HIPERBASES Y WWW

Se viene llamando *hiperbases* a aquellos sistemas que extraen información de una base de datos para generar hipermedia. En el caso de la WWW, las hiperbases existen gracias al interfaz CGI, que permite al cliente invocar a programas del servidor, en lugar de pedirle simplemente ficheros HTML estáticos. Esos programas, apoyándose eventualmente en el interfaz *Forms* para dialogar con el usuario cliente, le ofrecen la posibilidad de componer consultas a la base de datos y, tras interpretarlas y ejecutarlas, le devuelven los resultados en forma de documentos HTML dinámicos.

El problema de diseño de muchas hiperbases así concebidas consiste en que se necesita escribir los programas CGI manualmente: es evidente, pues, la conveniencia de una herramienta para generar automáticamente esos programas que exija al diseñador del sistema un conocimiento mínimo, incluso nulo, del interfaz CGI y del manejo del RDBMS (*Relational Data Base Management System*, sistema de administración de base de datos relacional) subyacente.

De las pasarelas entre RDBMSs y WWW publicadas hasta la fecha, tres se concentran en evitar la necesidad de programar, como lo hace CATALOG[1] (una herramienta de generación de hiperbases desarrollada siguiendo las ideas presentadas en esta ponencia):

- SWOOP[2] es una meta-herramienta: un entorno generador de aplicaciones CGI para consultar el contenido de una base de datos ORACLE. El diseño del sistema implica tener ciertos conocimientos de ORACLE y de HTML, ya que se debe especificar el comportamiento de los programas CGI mediante el uso de una sintaxis HTML extendida.
- MORE[2] resulta del enfoque inverso, puesto que no requiere programar, pero sí necesita de una ordenación de los contenidos de la base de datos para crear una meta-base de datos, que permite luego al sistema realizar búsquedas complejas y flexibles.
- WDB[4], aunque trabaja sobre bases de datos SyBase, parte de una idea muy similar a la de CATALOG, dado que logra su objetivo sin exigir al diseñador del sistema ni programación, ni reorganización de la base de datos.

GENERACIÓN DE PASARELAS

Una pasarela (*gateway*) a un sistema de almacenamiento se encarga de proporcionar la información al correspondiente servidor de información[5]. En el caso de WWW, el servidor de información se identifica con el servidor httpd que sirve información a los diversos clientes o navegadores.

En el caso de servir documentos estáticos la pasarela no existe, ya que el servidor accede directamente al sistema de ficheros. Sin embargo, en el caso de generación dinámica de documentos existe un programa encargado de interpretar la petición del cliente, obtener la información requerida, componer el documento de respuesta y pasar la información al servidor de información. Como se comentó en la sección anterior, en WWW la comunicación entre el servidor y la pasarela se lleva a cabo mediante el interfaz CGI. Cabe mencionar que, aunque la pasarela conste de diversos programas, ésta se considera única y compuesta por el conjunto de servicios proporcionados mediante los diversos programas.

Características

Las características generales de una pasarela, si bien no todas son necesarias, se pueden agrupar en:

- interpretación y formato: interpreta la petición enviada por el cliente y compone el documento de respuesta en el formato adecuado al cliente (HTML en el caso que se trata en esta ponencia).
- composición de documento: haciendo uso de las otras funcionalidades, se encarga de componer el documento de respuesta a la petición cursada por el cliente. Los documentos a componer pueden ser informativos solamente o interactivos: este último es el caso de documentos que contengan enlaces, formularios para consulta, …
- soporte de sesión: una de las mayores desventajas de WWW es la carencia del concepto de sesión, debido a la naturaleza *state-less* (sin estado) del protocolo http. Por ello, es muy interesante proporcionar mecanismos para el soporte de estado de la sesión dentro de la pasarela.
- control de acceso: si bien existen diversos mecanismos de control de acceso en el protocolo http, resulta interesante aumentar estos dentro de la pasarela. Esto permite poder utilizar la pasarela en otros servidores de información que carezcan de control de acceso, permitiendo además un filtrado de información (ver a continuación) integrado dentro de la pasarela.
- filtrado de información: la posibilidad de tener diferentes vistas de los contenidos del sistema de almacenamiento accedido requiere la existencia de mecanismos de filtrado de la información en la pasarela. Éste puede basarse en el nivel de acceso asignado por el proveedor de información mediante los mecanismos de control de acceso, o en las preferencias del usuario. Esta última opción hace necesaria la existencia de mecanismos (a través de documentos interactivos) mediante los cuales el usuario pueda indicar sus preferencias. Si existe soporte para sesión, las preferencias podrán mantenerse durante toda la sesión de acceso a la información, mientras que en el caso de no existir deberán indicarse en cada acceso a la pasarela.

Funciones

Para lograr el acceso al sistema de almacenamiento subyacente, la pasarela deberá llevar a cabo las siguientes funciones:

- definición de la estructura de consulta (campos de búsqueda, tipos, posibles combinaciones de consultas, …) y del mecanismo empleado, así como la adaptación al lenguaje de consulta soportado por el sistema de almacenamiento.
- definición de la estructura de los documentos a presentar, pudiendo combinar los diversos elementos de información disponibles en el sistema de almacenamiento. Esto incluye la definición de documentos haciendo uso de, por ejemplo, diversas tablas de una base de datos incluyendo además información externa a ésta almacenada en el sistema de ficheros.

- interpretación de las peticiones del usuario.
- composición de documentos (interactivos o no) que respondan a la consulta, teniendo en cuenta la información disponible para filtrado de información, formato de presentación (*layout*), datos de la sesión (por ejemplo, idioma elegido), …

A la hora de componer documentos, la pasarela debe tener conocimiento de la estructura del sistema de almacenamiento. Cada sistema de almacenamiento define intrínsecamente información sobre los tipos de datos de los componentes y la red de enlaces estructurales, esto es, enlaces intra- e inter-documentos.

Los enlaces intra-documentos son enlaces estructurales que definen las reglas de composición del sistema de almacenamiento, que pueden expresarse como una plantilla que contiene los enlaces a los componentes requeridos para componer el documento.

Los enlaces inter-documento definen la navegación, si no se modifica su naturaleza, o una posible estructura con una relación de contenido o jerarquía. Estos enlaces son los que quedan definidos por las restricciones referenciales en una base de datos relacional o la jerarquía de directorios de un sistema de ficheros.

Si bien esta plantilla de información, puede ser obtenida cada vez que se accede al sistema, la existencia de una copia física aumenta el rendimiento de la pasarela, a la vez de permitir la inclusión de *meta-datos*: nivel de permiso, agrupamiento para presentación, enlaces adicionales, inclusión de documentos relacionados mediante enlaces inter-documento, etiquetado en lenguaje natural, etc. Esta información adicional a la que se obtiene de la estructura del sistema de almacenamiento, permite aumentar las funcionalidades de la pasarela.

El concepto de plantilla de información sobre la estructura del sistema de almacenamiento y la edición de ésta para añadir meta-datos juega un papel fundamental a la hora de proporcionar funcionalidades a una pasarela como se verá en la descripción del sistema CATALOG, que es el tema del resto de la ponencia.

EL SISTEMA CATALOG

El sistema CATALOG es una pasarela a bases de datos relaciones, que junto con un servidor WWW se convierte en una hiperbase.

CATALOG opera sobre bases de datos ORACLE y permite al creador de la hiperbase ofrecer mecanismos de consulta vía WWW sin escribir una sola línea de código y sin modificar en nada la base de datos. Desde el punto de vista del diseñador del sistema, la pieza fundamental de CATALOG es el creador de plantillas: un programa que lee la estructura de la base de datos y elabora, para cada uno de sus usuarios, una plantilla maestra que puede ser modificada posteriormente para adecuar la presentación de las fichas (los documentos dinámicos generados por las consultas a la base de datos) a cada caso particular. La instalación de CATALOG no requiere más que la ejecución del creador de plantillas y la copia de los programas CGI, una vez compilados, al directorio correspondiente del servidor WWW.

Arquitectura software

Desde una óptica más interna, los servicios ofrecidos por CATALOG están estructurados en tres capas claramente separadas: el conjunto de interfaces CGI, el núcleo funcional, y el interfaz con la base de datos. Estas tres capas se muestran en la figura 1, y se detallan en las siguientes secciones.

Conjunto de interfaces CGI

El conjunto de interfaces CGI, propios de cada servicio, permite tener un control más flexible sobre la extracción de información de la base de datos. Aunque no todos los servicios son necesarios en una configuración concreta, en principio existen los siguientes:

- entries: lee el nombre (*login*) y clave (*password*) introducidos por el usuario en una *form* HTML y devuelve el conjunto de entradas que están disponibles para ese usuario.
- fields: devuelve los campos de búsqueda disponibles para un usuario y entrada dados.
- items: devuelve, para un conjunto de campos de búsqueda dado, un documento dinámico que permite realizar una consulta a la hiperbase. En función del tipo de los campos de búsqueda seleccionados en el documento generado por el servicio fields, habrá diversos tipos de elementos de búsqueda:
    - lista de términos seleccionables, si el tipo es WRD.
    - una ventana de búsqueda libre, si el tipo es FRE o TXT.
    - la lista de términos y la ventana de búsqueda libre, si el tipo es STR.

  Se pueden utilizar todos los elementos para componer la consulta, teniendo en cuenta que los términos pertenecientes al mismo campo se agrupan mediante una relación lógica AND y los de distinto campo mediante OR. Este mecanismo de consulta, si bien restringido, es sencillo y relativamente potente.

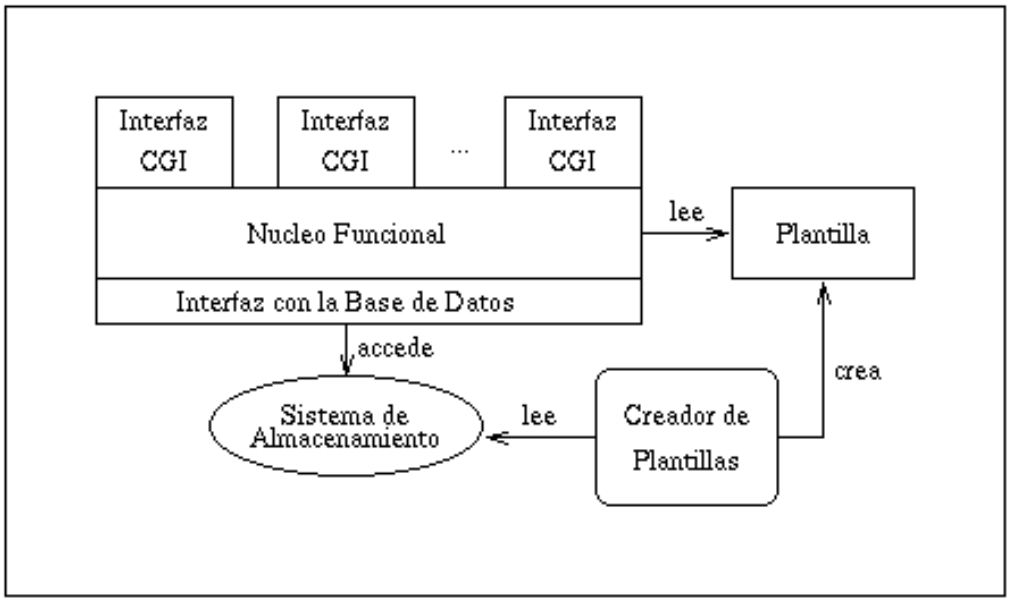

Figura 1. Módulos software de CATALOG

Esta descripción del servicio items se corresponde al *modo de consulta* de la hiperbase. Si por el contrario se elige el *modo de navegación* (i.e.: solamente habrá enlaces, y no formularios), este servicio devolverá un documento conteniendo una lista de términos que serán enlaces que ocultarán la consulta al usuario. Este modo, incrementando la sencillez de la búsqueda, limita la posibilidad de consultas más dirigidas y específicas, siendo más adecuado en el caso de hiperbases de tamaño reducido o tipo catálogo comercial, donde los documentos a localizar contienen relativamente pocos componentes de información.

- hits: devuelve el número de aciertos para una consulta dada. Si el número de aciertos es menor que una variable de entorno del sistema CATALOG (personalizable para cada instalación), se devuelven los descriptores de los documentos o fichas que cumplen la condición de búsqueda, mientras que si es mayor se devuelve una lista de selección (múltiple si se usa el modo de consulta) con las etiquetas de estos documentos.
- shorts: devuelve un conjunto de descriptores, cada uno con un enlace a la ficha correspondiente.
- cards: devuelve finalmente una ficha, es decir, el documento completo referente a una unidad de información del sistema de almacenamiento (base de datos).

Antes de crear el documento dinámico pertinente, cualquiera de los servicios CGI lee la plantilla correspondiente al usuario conectado para filtrar la información como proceda y para dar el formato adecuado a la página que será finalmente presentada. Esto se hace gracias al núcleo funcional.

.

Núcleo funcional

El núcleo funcional, compartido por todos los servicios, proporciona funcionalidades genéricas como, por ejemplo, las relacionadas con los permisos de acceso, el filtrado de información, la localización de los datos, el acceso a la información, la composición de los documentos, etc., y unas funcionalidades específicas (públicas, frente a las otras que son privadas) que proporcionan la función utilizada por los diversos servicios proporcionados mediante los interfaces CGI.

El formato de los documentos en HTML se encuentra separado del resto de código, lo cual permitirá la reutilización del resto del código para otros servidores de información que hagan uso de distintos formatos de documento.

Tras leer la plantilla (ver figura 1) y recuperar los componentes del documento, el núcleo funcional comprueba el nivel de permiso de cada componente con el nivel de acceso del usuario. Tras identificar los componentes accesibles para el usuario particular, se accede al sistema de almacenamiento mediante el interfaz con la base de datos para obtener los contenidos, y a continuación se compone el documento de acuerdo con la estructura (de consulta o documento) definida en la plantilla

Actualmente, el control de acceso está implementado mediante un sistema muy sencillo basado en el que UNIX tiene para los ficheros: cuando un usuario entra en el sistema a través del servicio CGI entries, el agente de seguridad verifica su identidad y recupera su perfil, que corresponde a un nivel de acceso, que es el que se usa para filtrar la información. Por otra parte, el perfil de usuario es la base para el establecimiento de un *ticket* de sesión, en el que también se incluye información sobre el idioma elegido para dicha sesión, y que se pasan unos CGIs a otros para mantener en lo posible una información de estado dentro de un sistema esencialmente *state-less* como es uno basado en el http.

Interfaz con la base de datos

El interfaz con la base de datos, también común a todos los servicios, garantiza que el núcleo funcional no tenga que preocuparse de qué RDBMS concreto exista por debajo, sino que pueda limitarse a traducir todas las consultas a un conjunto de llamadas SQL sencillas y fácilmente implementables en cualquier SQL-RDBMS.

Las funciones requeridas son:

- connect_db: conecta a un usuario a la base de datos.
- disconnect_db: lo desconecta.
- count_hits: devuelve el número de registros que cumplen una sentencia SELECT de SQL.
- get_hits: devuelve los valores correspondientes a un campo (columna) que cumplen la sentencia SELECT.
- get_hits_all: igual que el anterior, pero devuelve también valores vacíos, en el caso que el registro cumpla la condición de la sentencia SQL, pero no tenga ningún valor en la columna seleccionada.

Estás funciones han sido identificadas como el conjunto mínimo que permite consultar cualquier RDBMS. Por supuesto, el núcleo funcional debe encargarse de llevar a cabo consultas complejas haciendo uso de estas simples funciones.

En el caso de CATALOG, el RDBMS es el de Oracle, por lo que el interfaz con la base de datos se ha implementado con el precompilador Pro*C, aunque se prevé la posibilidad de adaptarlo a otros RDBMSs como los de INFORMIX (actualmente en curso de realización), SyBase e Ingres, usando los precompiladores de C correspondientes.

Plantillas

CATALOG hace uso de las plantillas para el filtrado de información y la composición de los documentos. Para cada RDBMS soportado debe existir un *creador de plantillas* (ver figura 1), que crea una *plantilla maestra* para usuarios de la base de datos. Esta plantilla maestra contiene todas las tablas, columnas y referencias cruzadas (restricciones referenciales) de ese usuario. Una vez que existe esa plantilla, los servicios de CATALOG (programas con interfaz CGI) se ejecutan correctamente, aunque la edición (personalización) de la plantilla permite hacer uso de todas las funcionalidades ofrecidas por CATALOG.

Sintaxis de la plantilla

Para cada tabla accesible por el usuario para el cual se creó la plantilla, se crea una entrada en la Plantilla Maestra con la siguiente sintaxis:

```
**
IU_REF,IU_link,permission_level,label
number_of_components
{component}
...
{component}
IU_END,IU_link
**
```

donde:

`IU_REF`: comienzo de una unidad de información (*Information Unit REFerence*).

`IU_link`: la clave primaria de la tabla y el nombre de la tabla separada por el carácter '~' (`clave_primaria~nombre_de_la_tabla`).

`permission_level`: un entero asignando un nivel mínimo de permiso para acceder la unidad de información o uno de los componentes.

`label`: una etiqueta para la unidad de información o un componente. El Generador de Plantilla crea las etiquetas a partir de los nombres de los componentes de la base de datos (cambiando el separador '_' , si existe, por un blanco). Si el componente es un enlace a un componente (ver el tipo REF:IU más adelante), la etiqueta consta de la palabra LINK seguida por el número de enlace.

`number_of_components`: el número de componentes de la unidad de información.

`{component}`:
`component_type,component_link,permission_level,reference_flag,search_flag,document_flag,descriptor_flag,label_flag,label`

`component_type`: unos de los posibles tipos descritos más adelante.

`component_link`: si el componente es una columna, el nombre de la columna; si corresponde a un enlace entre tablas (tipo REF:IU), la definición del enlace incluyendo el anclaje (el nombre de la columna en la tabla origen) y el destino (el nombre de columna y

de tabla destino), separados por el carácter '~' (nombre_columna~nombre_columna~nombre_tabla); si corresponde a un elemento de agrupamiento (LAB o BAL), un estilo de agrupamiento (ver más adelante).

reference_flag: I o R, para incluir los datos del componente o una referencia a ellos.

search_flag: SEARCH o NOSEARCH, en función de si el componente se debe utilizar para hacer búsquedas o no.

document_flag: DOC o NODOC, en función de si el componente se incluye o no en el documento (ficha).

descriptor_flag: SHORT o NOSHORT, en función de si el componente se incluye o no en el descriptor de la ficha.

label_flag: LABEL o NOLABEL, en función de si el componente se incluye o no en el etiqueta de la ficha.

IU_END: final de una unidad de información.

Tipos de componentes

Actualmente los tipos de componentes soportados por CATALOG son los siguientes:

WRD: componente de texto, que permite búsquedas por lista cerradas de términos.

FRE: datos textuales, que permiten búsqueda por texto libre.

STR: componente textual que permite búsqueda por lista de términos y texto libre.

TXT: datos textuales preformateados, que permiten búsqueda por texto libre.

NUM: componente numérico, actualmente sin soporte de consulta.

DAT: componente de fecha, actualmente sin soporte de consulta.

BIN: componente de datos binarios genéricos.

IMA: componente de tipo imagen.

VID: componente de tipo vídeo.

AUD: componente de tipo audio.

REF:IU: enlace de una unidad de información (uno de sus componentes) a otra unidad de información, que se crean cuando existen restricciones referenciales entre tablas.

REF:{BIN,TXT,IMA,VID,AUD}:
enlace a un componente de tipo binario, textual, imagen, vídeo, o audio.

URL: componente de tipo dirección WWW (Internet resource locator) que se presenta como un enlace.

LAB: marca de comienzo de agrupamiento..

BAL: marca de final de agrupamiento..

Estilos de agrupamiento

La estructura de presentación de información (*layout* del documento) se basa en tablas como se comenta en la siguiente sección. Sin embargo, se proporcionan diversos estilos de agrupamiento para la personalización de la presentación mediante la edición de la plantilla. Estos estilos son los siguientes:

DEF_LIST: Los componentes del grupo se presentan como una lista de definición, siendo la etiqueta el término y los datos la descripción. Este es el estilo por defecto.

ROW: Los componentes del grupo se presentan uno a uno en fila.

COL: Los componentes del grupo se presentan uno a uno en fila

NUM_LIST: Los componentes del grupo se presentan en una lista numerada.

LIST: Los componentes del grupo se presentan en una lista sin numerar.

Presentación de documento

La presentación del documento (*layout*) se basa en tablas, donde cada componente es un elemento de una lista de definición, la cual se sitúa dentro de un celda única de la tabla (existen pequeñas variantes de estilo entre los documentos y descriptores). Este es el estilo de formato por defecto, el cual puede ser modificado haciendo uso de agrupamientos, si bien la tabla sigue siendo la base. Por supuesto, los agrupamientos son recursivos, esto es, pueden combinarse para dar lugar a presentaciones de documento complejas.

Personalización de la plantilla

Mediante la edición de la plantilla se puede:

- Reorganizar los componentes dentro de una unidad de información.
- Agrupar componentes y etiquetas de grupo mediante el componente LAB.
- Personalizar la presentación del documento mediante el componente LAB haciendo uso de estilos de agrupamiento.
- Identificar tipos BIN como lo que son en cada caso concreto (imágenes, vídeos, audios, etc.).

- Identificar componentes textuales (STR) que realmente son referencias a componentes almacenados fuera del RDBMS (por ejemplo, la localización de ficheros de imágenes o vídeos), modificándolos a REF:IMA, REF:VID, etc.
- Poner diferentes niveles de permiso a los distintos componentes.
- Modificar las etiquetas por defecto, a etiquetas más descriptivas
- Añadir soporte *multi-lingual*. La implementación actual hace uso de la etiqueta del componente (`label`), donde se separan la traducción a los diversos idiomas mediante el carácter '*%*'. El orden de las diferentes etiquetas (su traducción), debe coincidir con el que se defina en el formulario de entrada al sistema. El primero es el usado por defecto en caso de no existir el solicitado.

Ejemplo de uso de CATALOG

Se presenta a continuación un pequeño ejemplo de uso de CATALOG, en el que se supone la existencia de un sistema de almacenamiento sencillo que consta de una base de datos con dos tablas, y un sistema de ficheros donde se almacenan imágenes.

Las dos tablas de la base de datos contienen los registros para personal de nuestro grupo y estudiantes realizando su proyecto fin de carrera, estando relacionadas mediante una restricción referencial dada por la relación tutor-alumno. Las tablas se muestran a continuación:

```
table gti_staff
   (
   Login VARCHAR2(8) PRIMARY KEY,
   Nombre VARCHAR2(30) NOT NULL,
   Papellido VARCHAR2(50) NOT NULL,
   Categoria VARCHAR2(30) NOT NULL,
   Fecha_ingreso DATE NULL,
   Fotito VARCHAR2(80) NULL,
   Foto VARCHAR2(80) NULL
   );

table gti_students
   (
   Login VARCHAR2(8) PRIMARY KEY,
   Nombre VARCHAR2(30) NOT NULL,
   Papellido VARCHAR2(50) NOT NULL,
   Fecha_ingreso DATE NOT NULL,
   Tutor VARCHAR2(8) CONSTRAINT
      login_inexistente
      REFERENCES GTI_STAFF(Login),
   Proyecto VARCHAR2(80) NULL,
   Nota_proyecto NUMBER NULL,
   Notas LONG NULL,
   Fotito VARCHAR2(80) NULL,
   Foto VARCHAR2(80) NULL
   );
```

Instalación de los programas

El primer paso es instalar los programas CGI de CATALOG, de acuerdo a la configuración del servidor WWW.

Formulario de acceso a la hiperbase

A continuación se debe proporcionar en alguna página un punto de entrada a la hiperbase, mediante un formulario similar al que se muestra a continuación:

```
<HR>
<h2>Acceso a la hiperbase de personal del
GTI</h2>
<HR>
<FORM ACTION="/cgi-bin/gti/entries"
      METHOD=POST>
<BR>
nombre <input type="text" name="login"><BR>
clave <input type="password"
name="password"><BR>
<HR>
Idioma
<select name="language" size=1>
<option value=0> English
<option value=1> Español
</select><br>
<HR>
<input type="submit" value="enviar">
<input type="reset" value="borrar" >
</FORM>
<HR>
```

Si bien no es necesario entrar al sistema mediante el servicio entries, es recomendable, ya que es la única forma de hacer uso de los mecanismos de filtrado de información, control de acceso y soporte multi-lingual de un modo completo.

Generación de Plantillas

Para las tablas definidas anteriormente el Generador de Plantillas crea la siguiente plantilla:

```
**
**
IU_REF,LOGIN~GTI_PPS,0,GTI PPS
11
STR,LOGIN,0,I,SEARCH,DOC,SHORT,LABEL,LOGIN
STR,NOMBRE,0,I,SEARCH,DOC,SHORT,NOLABEL,NOMBRE
STR,PAPELLIDO,0,I,SEARCH,DOC,SHORT,NOLABEL,PAP
       ELLIDO
DAT,FECHA_INGRESO,0,I,SEARCH,DOC,SHORT,NOLABEL
       ,FECHA INGRESO
STR,TUTOR,0,I,SEARCH,DOC,SHORT,NOLABEL,TUTOR
STR,PROYECTO,0,I,SEARCH,DOC,SHORT,NOLABEL,PROY
       ECTO
NUM,NOTA_PROYECTO,0,I,SEARCH,DOC,SHORT,NOLABEL
       ,NOTA PROYECTO
TXT,NOTAS,0,Y,SEARCH,DOC;SHORT,NOLABEL,NOTAS
STR,FOTITO,0,I,SEARCH,DOC,SHORT,NOLABEL,FOTITO
STR,FOTO,0,I,SEARCH,DOC,SHORT,NOLABEL,FOTO
REF:IU,TUTOR~LOGIN~GTI_STAFF,0,R,NOSEARCH,DOC,
       NOSHORT,NOLABEL,ENLACE1
END_IU_REF,LOGIN~GTI_PPS
**
**
**
IU_REF,LOGIN~GTI_STAFF,0,GTI STAFF
```

```
8
STR,LOGIN,0,I,SEARCH,DOC,SHORT,LABEL,LOGIN
STR,NOMBRE,0,I,SEARCH,DOC,SHORT,NOLABEL,NOMBRE
STR,PAPELLIDO,0,I,SEARCH,DOC,SHORT,NOLABEL,PAP
        ELLIDO
STR,CATEGORIA,0,I,SEARCH,DOC,SHORT,NOLABEL,CAT
        EGORIA
DAT,FECHA_INGRESO,0,I,SEARCH,DOC,SHORT,NOLABEL
        ,FECHA INGRESO
STR,FOTITO,0,I,SEARCH,DOC,SHORT,NOLABEL,FOTITO
STR,FOTO,0,I,SEARCH,DOC,SHORT,NOLABEL,FOTO
REF:IU,LOGIN~TUTOR~GTI_PPS,0,R,NOSEARCH,DOC,NO
        SHORT,NOLABEL,ENLACE2
END_IU_REF,LOGIN~GTI_STAFF
**
**
**
```

Personalización de plantillas

Mediante la personalización se puede modificar la presentación del documento, así como proporcionar mecanismo de filtrado de información. En la plantilla modificada que se muestra a continuación los cambios llevados a cabo han sido:

- identificar los campos que contienen los nombres de los ficheros de imágenes, como tales (pasándolos de STR a REF:IMA).
- proporcionar etiquetas en inglés y español.
- llevar a cabo reagrupamientos y modificaciones de presentación (modificando los valores por defecto de la presentación en etiqueta, descriptor y documento).
- activar dos niveles de permisos, dando soporte a un nivel público (nivel de permiso 0, para visitantes) y a otro restringido (nivel de permiso 10, para miembros del grupo). Para terminar la activación, deberán de existir al menos dos usuarios del sistema: uno *anónimo* con nivel de acceso 0, y otro *personal* con nivel 10.

```
**
**
IU_REF,LOGIN~GTI_PPS,0,STUDENTS%ESTUDIANTES
13
REF:IMA,FOTITO,0,I,NOSEARCH,NODOC,SHORT,NOLABE
        L,THUMBNAIL%FOTITO
REF:IMA,FOTO,0,I,NOSEARCH,DOC,NOSHORT,NOLABEL,
        PHOTO%FOTO
STR,LOGIN,0,I,SEARCH,DOC,SHORT,LABEL,LOGIN%LOG
        IN
STR,NOMBRE,0,I,SEARCH,DOC,SHORT,NOLABEL,NAME%N
        OMBRE
STR,PAPELLIDO,0,I,SEARCH,DOC,SHORT,NOLABEL,SUR
        NAME%APELLIDO
LAB,,0,I,NOSEARCH,DOC,SHORT,NOLABEL,MASTER
        THESIS%PROYECTO
STR,TUTOR,0,I,SEARCH,DOC,SHORT,NOLABEL,ADVISOR
        %TUTOR
STR,PROYECTO,0,I,SEARCH,DOC,SHORT,NOLABEL,TITL
        E%TITULO
DAT,FECHA_INGRESO,10,I,SEARCH,DOC,NOSHORT,NOLA
        BEL,STARTING DATE%FECHA DE COMIENZO
NUM,NOTA_PROYECTO,10,I,NOSEARCH,DOC,NOSHORT,NO
        LABEL,CALIFICATION%NOTA PROYECTO
BAL,,0,I,NOSEARCH,DOC,SHORT,NOLABEL,MASTER
        THESIS%PROYECTO
```

```
TXT,NOTAS,10,I,NOSEARCH,DOC,NOSHORT,NOLABEL,NO
        TES%NOTAS
REF:IU,TUTOR~LOGIN~GTI_STAFF,0,I,NOSEARCH,DOC,
        NOSHORT,NOLABEL,ADVISOR%TUTOR
END_IU_REF,LOGIN~GTI_PPS
**
**
**
IU_REF,LOGIN~GTI_STAFF,0,STAFF%PERSONAL
8
REF:IMA,FOTITO,0,I,NOSEARCH,DOC,SHORT,NOLABEL,
        THUMBNAIL%FOTITO
REF:IMA,FOTO,0,I,NOSEARCH,DOC,SHORT,NOLABEL,PH
        OTO%FOTO
STR,LOGIN,0,I,SEARCH,DOC,SHORT,LABEL,LOGIN%LOG
        IN
STR,NOMBRE,0,I,SEARCH,DOC,SHORT,NOLABEL,NAME%N
        OMBRE
STR,PAPELLIDO,0,I,SEARCH,DOC,SHORT,NOLABEL,SUR
        NAME%APELLIDO
STR,CATEGORIA,10,I,SEARCH,DOC,SHORT,NOLABEL,JO
        B%CATEGORIA
DAT,FECHA_INGRESO,10,I,SEARCH,DOC,NOSHORT,NOLA
        BEL,STARTING DATE%FECHA INGRESO
REF:IU,LOGIN~TUTOR~GTI_PPS,0,R,NOSEARCH,DOC,NO
        SHORT,NOLABEL,STUDENTS%PFCs
END_IU_REF,LOGIN~GTI_STAFF
**
**
**
```

Resumen de las facilidades ofrecidas por CATALOG

Volviendo al resultado final observable por el diseñador de la hiperbase generada por CATALOG, las ventajas del sistema son las siguientes:

- Facilidad de instalación e inmediatez de uso: no es necesaria ninguna modificación del contenido de la base de datos que se quiere dar a conocer a través de Internet (el creador de plantillas se encarga de averiguar su estructura y de generar las plantillas maestras); tampoco hace falta escribir una sola línea de código: sólo copiar los programas, una vez compilados (si esto es necesario), al directorio adecuado.
- Personalización de las plantillas: opcionalmente, se pueden modificar las plantillas creadas automáticamente para reordenar los componentes dentro de las fichas, para cambiar permisos de acceso, para sustituir etiquetas generadas automáticamente por otras más explicativas, quizá en varios idiomas, etc.
- Filtrado dinámico de la información: con independencia de la posible existencia de otros mecanismos de control de acceso impuestos por el proveedor del servicio, CATALOG permite controlar qué información, de la contenida en la base de datos, es visible por qué usuario, según sea el nivel de acceso de su perfil.
- Creación de páginas en varios idiomas: CATALOG proporciona los mecanismos necesarios para la traducción de los títulos y etiquetas de las páginas

que serán generadas dinámicamente, siendo responsabilidad del diseñador del sistema su eventual inclusión en la hiperbase.

Situación actual

Actualmente existe un prototipo de CATALOG con el que se pueden consultar las bases de datos de nuestro Grupo y del Museo Arqueológico Nacional, cuyos URLs respectivos son http://www.gti.ssr.upm.es./ y http://www.gti.ssr.upm.es./~man/.

CONCLUSIONES

En esta ponencia se ha presentado la ventaja que supone la utilización de hiperbases para dar soporte a la provisión de contenidos en Internet. Tras introducir las ideas generales para dar soporte a hiperbases en WWW (una pasarela que dialoga con el servidor mediante el interfaz CGI y proporciona información generada dinámicamente), se presentan las características y funciones de cualquier pasarela a sistemas de almacenamiento.

Así mismo, se describe una pasarela a bases de datos relacionales, CATALOG, que permite la construcción sencilla de hiperbases en Internet sin tener que modificar la base de datos y sin programar una sola línea de código. Mediante la edición de una plantilla generada automáticamente por el sistema CATALOG, se puede personalizar la hiperbase (por ejemplo, modificando presentación de los documentos generados, sin tener que variar la estructura de las tablas de la base de datos) y añadirle funcionalidades de filtrado de información en función del tipo de usuario, soporte de etiquetas en varios idiomas, etc.

Finalmente se presenta un ejemplo sencillo de instalación y uso de CATALOG.

CATALOG se encuentra actualmente en desarrollo, existiendo algunas instalaciones del prototipo.

REFERENCIAS

[1] J. M. Martínez, y F. Morán, "CATALOG: A WWW gateway for RDBMS", WebNet'96, H. Maurer (de.), AACE 1996., pp. 342-347. *World Conference of the Web Society, San Francisco, Octubre 1996.*

[2] A. Hunter, R.I. Ferguson, S.Hedges, "SWOOP: An application generator for ORACLE/WWW system", http://www.w3.org/pub/Conference/WWW4/Papers/207/

[3] D. Eichmannm T. McGregor, D. Danle, "Integrating structured databases into the Web: The MORE system", *Computer Networks and ISDN Systems* 27 (1994), pp. 281-288.

[4] B.F. Rasmussen, "WDB - A Web interface to SQL Databases", http://arch-http.hq.eso.org/bfrasmus/wdb/intro.html

[5] J.M. Martínez, J. Bescós, y G. Cisneros, "Generic Information Engine for Use in Information System Servers", en *Multimedia Storage and Archiving Systems*, SPIE Proceeding Vol. 2916 (en prensa, 12 pp.), Boston, Noviembre 1996.