

UNIVERSIDAD POLITÉCNICA DE MADRID

FACULTAD DE INFORMÁTICA

TRABAJO FIN DE CARRERA

CoLogNetWS:

**Desarrollo de un Sitio Web con Actualización
Distribuída de las Bases de Datos en varios nodos**

Autor: Jorge A. Navas Laserna

Tutor: Daniel Cabeza Gras

Cotutor: Manuel V. Hermenegildo Salinas

Dedicado a mi madre y a mi padre.

Agradecimientos

En primer lugar me gustaría agradecer a toda mi familia su cariño y apoyo que cada día me demuestran. En especial, a mis padres y a mi hermano que siempre están cuando los necesito y respetan mis decisiones en cada momento. Entre todos hemos conseguido lo que nos propusimos hace ya algunos años, superando todos los obstáculos. Sin vuestra ayuda hubiera sido imposible.

También agradezco el apoyo incondicional de mis amig@s, un pilar en mi vida, que siempre han demostrado que me quieren y han estado junto a mí en mis peores momentos. Soy un privilegiado siendo vuestro amigo. En especial, quiero agradecer a Jorge, Jose, Lili, Manolo, Natalia y Raquel su ayuda constante durante todo este tiempo.

Por supuesto, quiero agradecer a todo el CLIP el magnífico trato recibido, todos los conocimientos que he adquirido gracias a ellos durante el desarrollo de este proyecto y, cómo no, que hayan sido capaces de aguantarme en todo momento. En especial, quiero agradecer a Jose M. toda su ayuda ya que gran parte de lo conseguido es gracias a él, y a Astrid que tantos problemas me ha solucionado.

No quisiera olvidarme de mis compañer@s de facultad, agradeciéndoles toda su ayuda, ya que esto es parte de todos, y es un colofón a todos los años de carrera que hemos compartido, en los que hemos sufrido y en el que también hemos disfrutado mucho. No os olvidaré.

Jorge A. Navas Laserna

Universidad Politécnica de Madrid

Facultad de Informática

Índice general

Agradecimientos	I
Índice general	III
Índice de figuras	V
Resumen	VII
1.. Introducción	1
2.. Objetivos	3
3.. Antecedentes de CoLogNetWS	7
3.1. El protocolo HTTP	7
3.2. La librería PiLLoW	11
3.3. Librería de predicados persistentes: persdb	12
3.4. RadioWeb: Generación automática de sitios Web	14
3.5. WebDB: Un interfaz WWW Genérico a Bases de Datos	14
3.6. El proyecto eBroker	15
4.. La aplicación CoLogNetWS	17
4.1. Introducción	17
4.2. Entornos de desarrollo y aplicación de CoLogNetWS	18

4.3.	Esquema de la arquitectura de la aplicación	18
4.3.1.	Interfaz WWW	20
4.3.2.	Interfaz inter-nodos	24
4.3.3.	Entidad	25
4.3.4.	Base de datos	25
4.4.	Los servicios de CoLogNetWS	26
4.4.1.	Visualizar/Insertar/Modificar/Eliminar a través del interfaz WWW	26
4.4.2.	Envío y Recepción de datos para la sincronización de las bases de datos	30
4.4.3.	Insertar/Modificar/Eliminar a través del interfaz XML	33
4.4.4.	Anotación / Consulta de las transacciones remotas	33
5..	Conclusiones y Trabajo futuro	36
5.1.	Conclusiones: Consideraciones finales sobre CoLogNetWS	36
5.2.	Trabajo futuro	38
	Bibliografía	41
	A..Manual de referencia de CoLogNetWS	43

Índice de figuras

3.1. El Interfaz CGI	9
3.2. Arquitectura de comunicaciones de CoLogNetWS	10
4.1. Esquema general de la arquitectura de CoLogNetWS	19
4.2. Operaciones desde WWW	29
4.3. Envío y Recepción de datos a/desde otros sitios Web	32
4.4. Operaciones desde otros sitios Web	34
4.5. Anotación / Consulta de las transacciones remotas	35

Resumen

El objetivo fundamental que se pretende alcanzar es la creación de un sitio Web y la definición de un protocolo de intercambio de datos. Para ello, la herramienta desarrollada, CoLogNetWS, acrónimo de CoLogNet Web-site, cumple las siguientes características:

- En primer lugar, permite acceder o modificar, a través de cualquier navegador WWW, la información almacenada en el sistema.
- Por otro lado, permite la comunicación con otros sitios Web, a través de la definición e implementación de un protocolo de intercambio de datos.
- Alberga una base de datos local, cuyas tablas están basadas en predicados persistentes Prolog.
- Cuenta con un interfaz de usuario WWW reconfigurable en el que se puede modificar el entorno gráfico del sistema.
- Asimismo, cumple con una serie de requisitos de seguridad, mediante la utilización de vistas diferentes para el administrador del sistema y los usuarios normales, y de rendimiento, minimizando el tiempo de respuesta en las transacciones.

El desarrollo del proyecto ha constado de varias fases. En primer lugar se desarrolló las funcionalidades básicas del sitio Web, y se definió el interfaz de usuario WWW. Posteriormente se desarrolló el intercambio de información entre los sitios Web y, finalmente, se fueron añadiendo otras funcionalidades al sistema.

Capítulo 1

Introducción

Hoy en día, los términos Internet y WWW son ampliamente conocidos debido al crecimiento exponencial que la actividad en la red ha experimentado en los últimos años. Mucha gente depende de las aplicaciones por Internet, tales como correo electrónico o acceso Web. Por todo ello, en estos últimos tiempos, gran cantidad de esfuerzo en el campo de la informática ha estado encaminado hacia el desarrollo de las tecnologías Web.

Por otra parte, el desarrollo de aplicaciones estándar que ofrezcan un buen soporte en lo referente a conectividad entre redes y protocolos de comunicación no siempre dan el resultado esperado. En muchas ocasiones encontramos problemas que podrían ser resueltos de una manera mucho más eficaz si se aprovecharan las ventajas que ofrece la Programación Lógica, como el procesamiento simbólico, gramáticas, bases de datos dinámicas, facilidades de búsqueda, o meta-programación.

Con el desarrollo de la aplicación CoLogNetWS se demuestra una vez más cómo la Programación Lógica es una opción cada vez más en alza para el desarrollo de aplicaciones Web, siendo en muchos casos la más idónea debido al alto nivel de abstracción ofrecido por Prolog.

Si dejamos a un lado las capacidades particulares de la Programación Lógica y nos fijamos sólo en aquellas que se le piden a otros lenguajes propuestos recientemente como herramientas de programación de red, como es el caso de Java y sus derivados, vemos como las actuales implementaciones de Prolog también cuentan con

manipulación de memoria dinámica, otorgan a sus desarrollos robustez y posibilitan la generación de código de byte independiente de la arquitectura.

Como respuesta básica a esta demanda de aplicaciones de red, la mayoría de los lenguajes LP/CLP cuentan con una serie de recursos al nivel del protocolo TCP/IP. Entre ellos podemos destacar, a modo de ejemplo, la comunicación entre procesos vía sockets.

Sin embargo, si queremos trabajar con el protocolo HTTP en el ámbito de la programación WWW, necesitamos algo más, una capa superior en la pila de protocolos de la arquitectura TCP/IP. En el sistema Ciao Prolog, como se describe en [CHV99], esto se ha resuelto mediante el desarrollo de la librería *PiLLoW* que simplifica el proceso de generación de este lenguaje.

No obstante, el nivel de desarrollo del sistema Ciao Prolog en cuanto a aplicaciones Web ha seguido progresando. Puesto que una parte básica a la hora de hacer cualquier aplicación Web es el almacenamiento de los datos, el sistema Ciao Prolog ofrece también una solución de fácil uso mediante predicados persistentes o bien a través de accesos a bases de datos relacionales remotas o locales. Estos servicios son ofrecidos por las librerías *persdb* [GCH98a] y *persdb_sql* [CCG⁺98].

La última aportación de Ciao Prolog al desarrollo Web es la manipulación de ficheros XML, mediante una extensión del lenguaje que permite realizar consultas y búsquedas eficientes sobre un documento XML. Esta tarea forma parte de las líneas de investigación actuales del grupo de Computación Lógica, Implementación y Paralelismo (CLIP) de la Universidad Politécnica de Madrid.

Capítulo 2

Objetivos

En este capítulo explicaremos qué problemas queremos resolver mediante este desarrollo y cuáles son los requisitos que debe cumplir. Este punto sienta las bases de la aplicación y permite una mejor comprensión del resto del documento.

CoLogNetWS surge para paliar la necesidad de crear un sitio Web donde estén contenidos todos los aspectos relacionados con la Programación Lógica estándar y con restricciones (LP/CLP). Esta necesidad no se entiende en toda su totalidad si no es dentro del marco de CoLogNet.

CoLogNet es la Red de Excelencia dedicada a la Lógica Computacional. CoLogNet tiene un ámbito mayor que su red hermana en los Estados Unidos, llamada Compulog Americas, ya que pretende abarcar todos los campos de la Lógica Computacional y Aplicaciones Lógicas. Su objetivo principal es unificar e integrar a estas subcomunidades dispersas. Además de esta máxima, el principio fundamental de CoLogNet es promover el intercambio y la cooperación entre las diferentes áreas que lo componen. Este objetivo determina gran parte de los requisitos que tiene que afrontar la aplicación.

CoLogNet está dividida en una serie de áreas y define un nodo para cada una de ellas:

- Área 1 - Metodologías lógicas.
- Área 2 - Programación lógica estándar y con restricciones.

- Área 3 - Entornos y sistemas de lógica computacional.
- Área 4 - Información basada en la lógica y sistemas de conocimiento.
- Área 5 - Lógica y sistemas multi-agentes.
- Área 6 - Procesamiento de lenguaje natural.
- Área 7 - Métodos formales, especificación y verificación.
- Área 8 - Razonamiento automático, deducción, demostración de teoremas y modelo de comprobación.
- Área 9 - Ingeniería eléctrica y mecánica en la lógica.

La importancia de la aplicación que se ha desarrollado en este trabajo radica en que constituye un nodo de la red de CoLogNet, ocupándose del área 3, y además establece el protocolo de intercambio de información que permite la cooperación entre los distintos nodos. Por ello, se trata de un proyecto que surge como respuesta concreta a una necesidad real. A continuación se presentarán las principales funciones que el sistema debe realizar, la información que se utiliza, las restricciones y otros factores que afectan al desarrollo del mismo.

- Gestión de las diferentes entidades de las que se guarda informacin:
 - Gestión de los grupos de investigación.

Se requiere gestionar la siguiente información de un grupo: dirección Web y nombre completo y abreviado tanto del grupo como de la institución a la que pertenece; dirección, código postal, ciudad y país donde se encuentra. Además de la información propia del grupo se tratarán también sus miembros, personas de contacto del grupo, proyectos desarrollados, aplicaciones informáticas realizadas, y las áreas en la que investiga el grupo.

- Gestión de los investigadores individuales o de grupos.

De un investigador tendremos que almacenar: nombre y apellidos, título, dirección de su página Web, email, teléfono, fax, descripción, y sus datos de localización (dirección, código postal, ciudad y país). Igualmente se gestionará los grupos con los que colabora, los grupos de los que es contacto, y las aplicaciones informáticas de las que es contacto.

- Gestión de los proyectos de investigación.

A la hora de gestionar los proyectos realizados se tendrá que almacenar: título completo y abreviado; dirección Web, categoría; descripción y comentarios adicionales; y finalmente los grupos que lo gestionan.

- Gestión de los entornos y aplicaciones informáticas.

Se tendrá que almacenar: nombre completo y abreviado de la aplicación; dirección Web y dirección FTP; plataformas que la soportan, licencia y breve descripción. Como información adicional se guardarán los grupos que han desarrollado la aplicación y las personas de contacto.

- Gestión de las áreas de investigación.

En cuanto a una área, se quiere almacenar su nombre completo y abreviado además de una breve descripción. Adicionalmente, se requiere gestionar los grupos que se clasifican dentro de dicha área.

- Gestión de los principales eventos.

Se deberá gestionar de un evento: título, categoría, fecha de inicio del evento, fecha de finalización, fecha límite de suscripción, comentarios adicionales y email de una persona de contacto.

- La aplicación permitirá operar sobre el sistema a través de cualquier navegador WWW, ofreciendo las siguientes operaciones:

- Visualización del contenido de un objeto perteneciente a una determinada entidad almacenada en el sistema.
 - Inserción de un objeto perteneciente a una determinada entidad en el sistema.
 - Modificación del contenido de un objeto perteneciente a una determinada entidad.
 - Eliminación de un objeto perteneciente a una determinada entidad.
-
- Ofrecerá un GUI (Graphic User Interface) intuitivo y fácilmente reconfigurable.
 - Definirá e implementará un protocolo de comunicación que permita el intercambio de información con otros sitios Web de la red de CoLogNet.
 - Permitirá definir dos tipos de usuarios: usuario normal y administrador. El primero tendrá un acceso restringido al sitio Web, mientras que el administrador en cambio tendrá todos los privilegios. Adicionalmente, el administrador podrá comprobar el resultado de los intercambios realizados con otros sitios Web.

Capítulo 3

Antecedentes de CoLogNetWS

CoLogNetWS es una aplicación que se fundamenta sobre el protocolo HTTP; todos los servicios que ofrece se proveen a través de este protocolo. La aplicación ha sido desarrollada con el sistema de Programación Lógica Ciao Prolog, haciendo un uso intensivo de sus librerías `PiLLoW` y `persdb`. Cabe destacar la influencia que otros proyectos han tenido sobre CoLogNetWS, como RadioWeb, WebDB y eBroker.

En este capítulo intentaremos facilitar un acercamiento a cada uno de ellos, sin entrar en demasiados detalles, de forma que quede claro en qué consisten, así como lo que aportan a este problema particular.

Para mayor profundidad, se pueden consultar los documentos sobre el protocolo HTTP [BLFIF96], `PiLLoW` [CHV99], `persdb` [GCH98a], RadioWeb [CHGT98] y WebDB [GCH98b].

3.1. El protocolo HTTP

El protocolo HTTP (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP.

En Internet, la mayoría de las aplicaciones, y en concreto el protocolo HTTP, se desarrollan bajo el concepto del modelo cliente-servidor, aunque actualmente está tomando mayor importancia el desarrollo de aplicaciones basadas en agentes móviles. El paradigma cliente-servidor es un modelo arquitectónico de sistemas informáticos

basados en redes que permite que los datos se almacenen en un formato estándar en las máquinas servidoras para ponerlos a disposición de usuarios que podrán acceder desde clientes remotos, independientemente de la configuración de estos últimos, que seguramente será diferente en la mayor parte de los casos.

Desde el punto de vista de las comunicaciones, el protocolo HTTP está soportado sobre los servicios de conexión TCP/IP, y funciona de la misma forma que el resto de servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP, y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor enviando un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actuar; cada objeto Web (documento HTML, fichero multimedia, aplicación CGI,etc) es conocido por su URL.

Las principales características del protocolo HTTP son:

- Permite transmitir cualquier tipo de documento respetando su formato original.
- Permite la transferencia de objetos multimedia.
- Existen tres operaciones básicas: GET, para pedir un objeto; POST, para enviar información al servidor y HEAD, para solicitar las características de un objeto.
- Cada operación HTTP implica una conexión con el servidor, que es liberada al término de la misma. Es decir, en una operación se puede recoger un único objeto.

- No mantiene estado.
- Cada objeto se identifica a partir de su URL.

La forma más sencilla de realizar aplicaciones WWW es a través del protocolo conocido como **Common Gateway Interface (CGI)**. Un ejecutable CGI se distingue de los ejecutables normales en que el resultado de su ejecución, su salida, consiste en algo que el navegador WWW que esté siendo utilizado pueda entender, es decir, generalmente código HTML.

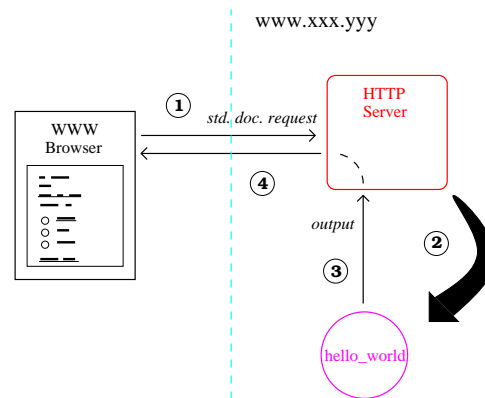


Figura 3.1: El Interfaz CGI

La idea principal de cómo funciona un ejecutable CGI se puede observar en la Figura 3.1. Básicamente, cuando un usuario se conecta a la dirección de un ejecutable CGI en un documento (por ejemplo, `http://www.xxx.yyy/foo/hello-world.cgi`), el proceso es el siguiente:

El servidor HTTP reconoce que no se trata de un documento normal sino de un ejecutable CGI, bien porque está situado en un directorio especial llamado normalmente `cgi-bin`, o bien por su extensión, típicamente `cgi` (1). A continuación, el servidor HTTP arranca el ejecutable (2). Durante la ejecución se genera la salida (3), que debería estar en un formato que el navegador pueda entender, como HTML, y es volcada al navegador como si hubiera accedido a una página normal (4).

Basándonos en esta idea, la aplicación CoLogNetWS ha sido desarrollada como un conjunto de aplicaciones CGI implementadas como ejecutables Ciao que realizan operaciones a partir de peticiones realizadas por los clientes.

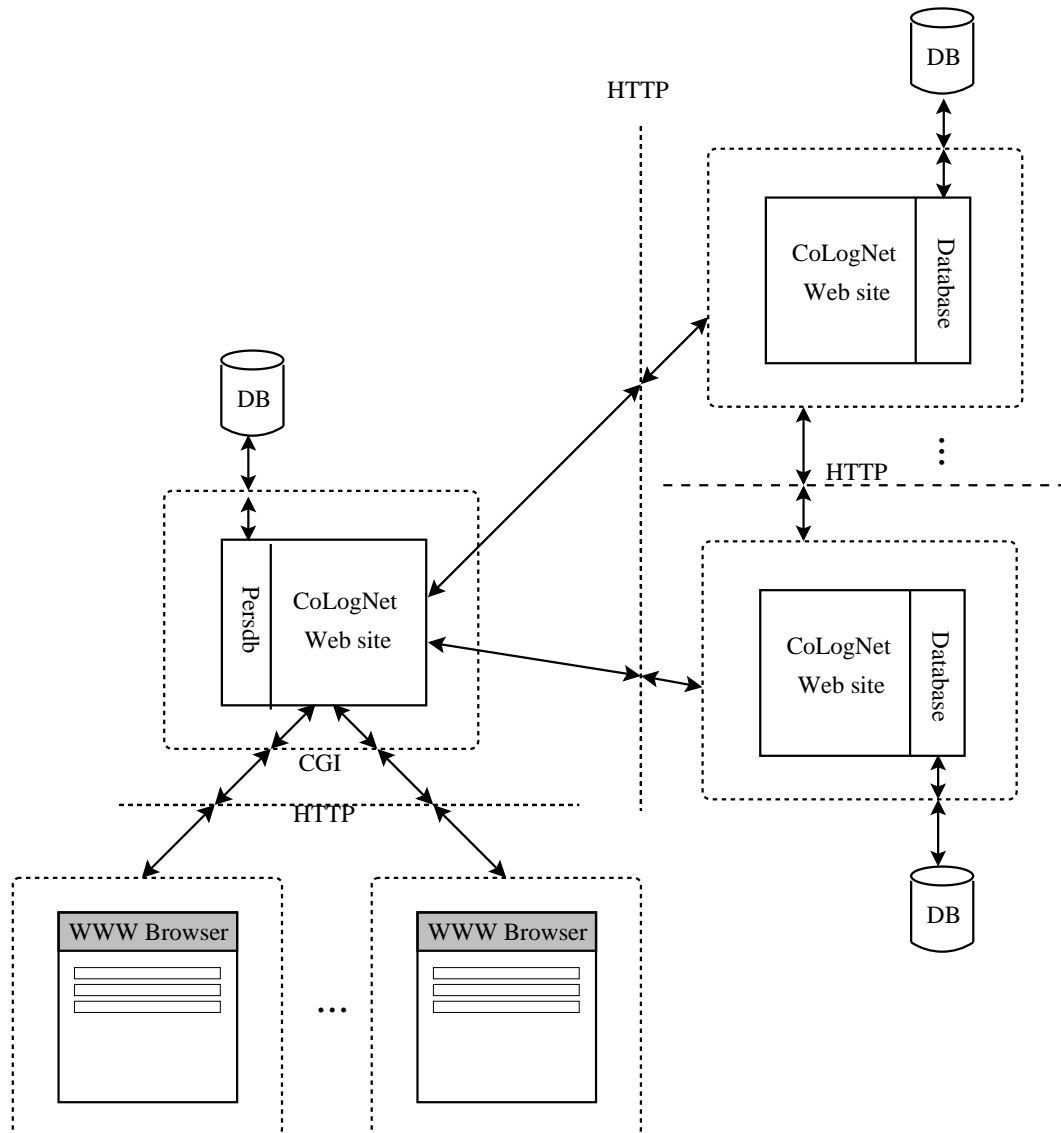


Figura 3.2: Arquitectura de comunicaciones de CoLogNetWS

En la Figura 3.2 se puede observar una representación de la comunicación existente entre los clientes WWW y el servidor CoLogNetWS, además de la comunicación

entre CoLogNetWS y el resto de sitios Web que actúan en ese caso tanto de cliente como de servidor. En ambos casos, esa comunicación se realiza a través del protocolo HTTP.

3.2. La librería *PiLLoW*

PiLLoW es una librería Prolog de dominio público diseñada para simplificar la programación de aplicaciones WWW e Internet en sistemas LP/CLP. *PiLLoW* ofrece facilidades para acceso a documentos y código de la WWW; parsing, manipulación y generación de documentos tanto HTML como XML; generación de formularios HTML; manejadores para scripts CGI y procesamiento de plantillas HTML/XML. Una importante contribución de *PiLLoW* es la modelización de código HTML/XML como términos Herbrand. La librería ha sido desarrollada en el contexto del sistema Ciao Prolog, aunque ha sido adaptada satisfactoriamente a numerosos sistemas LP/CLP.

A continuación se detallan las funcionalidades más significativas de *PiLLoW* :

- Generación de documentos HTML y XML estructurados que maneja como términos de Herbrand. Puesto que los sistemas LP/CLP desarrollan procesamiento simbólico usando términos de Herbrand, *PiLLoW* permite manejar código HTML y XML como términos. Por lo tanto, tales estructuras sólo necesitan ser traducidas por los predicados apropiados a código HTML/XML para mostrar la salida. En general, esta relación entre el código HTML/XML y los términos Prolog permite ver una página HTML o un documento XML como un término Herbrand.
- Creación y manejo de formularios HTML. Esta librería ofrece una serie de estructuras que permiten representar varios elementos relacionados con los for-

mularios HTML. Además provee algunos predicados que realizan y simplifican la tarea de escribir un programa Prolog que actúe como aplicación CGI. Básicamente facilita la traducción de datos de formularios HTML a un diccionario.

- Acceso a documentos WWW. Permite el acceso a documentos WWW, facilitando el desarrollo de aplicaciones tales como buscadores o analizadores de contenido.
- Plantillas HTML/XML. Las plantillas son ficheros que contienen código HTML/XML estándar, pero en los cuales se pueden poner huecos identificados a través de una etiqueta especial. Una de las ventajas principales del uso de plantillas es que permiten definir interfaces gráficas fácilmente reconfigurables.

En el desarrollo de CoLogNetWS, todo el manejo de documentos HTML/XML se ha realizado utilizando esta librería. Además la apariencia de la interfaz gráfica está desarrollada mediante plantillas HTML.

3.3. Librería de predicados persistentes: persdb

Esta librería implementa el concepto de predicados persistentes. Un predicado persistente es un tipo especial de predicado dinámico que reside físicamente en algún medio de almacenamiento, como un conjunto de ficheros.

La característica esencial de estos predicados es que los cambios realizados en ellos sobreviven entre diferentes ejecuciones del programa que los utilice. Es decir, si se interrumpe la ejecución del programa y se vuelve a arrancar, el predicado que el nuevo proceso ve está precisamente en el mismo estado en el que estaba en el momento de que la ejecución anterior fuera interrumpida. El uso de los predicados persistentes no difiere de la manera en que se utilizan los predicados dinámicos de Prolog.

Para efectuar cambios sobre los contenidos de estos predicados se utilizan primitivas análogas a las utilizadas con los predicados dinámicos. Estas actualizaciones son transaccionales, de modo que garantizan la seguridad ante una eventual pérdida de datos, como por ejemplo una caída del sistema. Además, gracias a la atomicidad de las transacciones, los predicados persistentes permiten actualizaciones concurrentes desde distintos programas.

Un rasgo muy interesante de esta noción de predicados persistentes es que se abstrae dónde está almacenado el predicado realmente. Así, un programa puede usar predicados persistentes almacenados en ficheros o en bases de datos relacionales externas indistintamente. Además el tipo de almacenamiento de un predicado puede modificarse sin que para ello haya que cambiar el programa.

Persdb ofrece una versión de predicados persistentes que permite implementar una forma de base de datos relacional a la vez simple y potente que no requiere soporte externo, aparte del mantenimiento del sistema de ficheros en el que se apoya. Esto se debe al hecho de que este tipo de predicados persistentes son almacenados en uno o más ficheros auxiliares en un mismo directorio.

Este tipo de bases de datos es especialmente útil para construir aplicaciones Prolog de tamaño moderado que requieran almacenamiento persistente. En muchos casos ofrece un modo mucho más sencillo de implementar dicho almacenamiento que utilizando ficheros bajo el control directo del programa, además de las facilidades que ofrece la librería para modificar los datos contenidos en estos predicados persistentes.

En el desarrollo de CoLogNetWS, todo el almacenamiento persistente (base de datos y predicados relacionados con los modos de acceso) se ha implementado a través de esta librería.

3.4. RadioWeb: Generación automática de sitios Web

RadioWeb fue creado para desarrollar y consecuentemente aplicar una tecnología para la generación automática de sitios Web, así como la gestión de sitios Web dinámicos cumpliendo los objetivos de reducción de costes de gestión y mantenimiento del sitio Web, e incremento de la calidad del sitio Web generado.

RadioWeb se basa fundamentalmente en:

- Almacenamiento de datos que pueden estar contenidos en sistemas de bases de datos relacionales (SQL) remotos o locales. Provee una interfaz genérica de acceso a los datos, que oculta los detalles de bajo nivel.
- Intérprete que posee un conjunto de reglas que permite proyectar descripciones abstractas (texto, imágenes,...) a descripciones de estructuras lógicas de esos objetos.
- Intérprete que traduce esas descripciones lógicas a un lenguaje entendido por el navegador Web, que generalmente es HTML.

CoLogNetWS toma de RadioWeb su forma de almacenamiento de datos local y la idea de separar el nivel lógico del nivel de aplicación.

3.5. WebDB: Un interfaz WWW Genérico a Bases de Datos

WebDB es un gestor de bases de datos accesible mediante el protocolo HTTP y reconfigurable en dos aspectos: las propias tablas y el interfaz gráfico de usuario.

Sus características principales son las siguientes:

- Permite navegar y modificar de manera interactiva y remota, a través de cualquier navegador WWW, los datos contenidos en multitud de sistemas de bases

de datos relacionales (SQL). Por tanto, es capaz de enlazar bases de datos SQL remotas de manera automática.

- También alberga bases de datos locales, cuyas tablas están basadas en predicados persistentes Prolog, permitiendo su mantenimiento, edición, creación y borrado.
- Ofrece un interfaz de acceso común a ambos tipos de datos, locales y remotos, independientemente de su situación geográfica.
- Cuenta con un interfaz de usuario WWW reconfigurable en el que se pueden modificar tanto el entorno gráfico del sistema como la representación de los campos de las tuplas de las tablas de la base de datos y los mensajes generados por el sistema para interactuar con el usuario.
- Cumple con una serie de requisitos de seguridad, mediante la utilización de vistas diferentes para el administrador del sistema y los usuarios normales.

CoLogNetWS toma de WebDB su medio de almacenamiento de datos locales y sus buenas cualidades de fácil adaptación del interfaz gráfico, así como el tratamiento de los distintos modos de acceso.

3.6. El proyecto eBroker

El proyecto eBroker nace como respuesta a una necesidad comercial, cuyo objetivo es que el usuario pueda definir una oferta o demanda sobre un determinado producto (categoría), de manera que el sistema case esa oferta o demanda con el resto de ofertas y demandas almacenadas en él, de la mejor manera posible. Estas ofertas y demandas se representan mediante ficheros XML acordes con la definición XDR de la categoría.

eBroker aporta las siguientes funcionalidades:

- Desarrolla un lenguaje básico de consultas y búsquedas sobre documentos o fragmentos XML, utilizando expresiones CLP, con el fin de poder la búsqueda y poder definir unas consultas más flexibles.
- Desarrolla un lenguaje de usuario sobre el anterior más intuitivo.
- Añade funcionalidades de ordenación, cardinalidad e índice al lenguaje anterior.
- Añade indexing para potenciar la búsqueda.
- Añade medios de validación semántica de un documento XML con respecto a un XDR dado.

La aplicación CoLogNetWS se inspira en algunas características de eBroker para el manejo y procesamiento de ficheros XML.

Capítulo 4

La aplicación CoLogNetWS

4.1. Introducción

En este capítulo se pretende dar una visión global de CoLogNetWS y de su implementación. Para ello, en primer lugar se hará una descripción del entorno de desarrollo de CoLogNetWS. En este punto hablaremos acerca de los sistemas operativos sobre los que se apoya y describiremos el lenguaje de programación utilizado, así como de los puntos relevantes de este lenguaje en particular y las facilidades que ofrece para el desarrollo de aplicaciones como ésta.

A continuación, intentaremos comprender mejor la estructura en la que se divide CoLogNetWS. Para ello, veremos un esquema de su arquitectura en el que se podrán observar los módulos o subsistemas que lo forman. Describiremos estos módulos, así como las interacciones presentes entre ellos.

Finalmente, describiremos los servicios ofrecidos por CoLogNetWS, comentando el interfaz de usuario y entrando en descripciones procedimentales y soluciones de implementación para cada uno de ellos.

En este documento se incluye el manual de referencia de la aplicación [NCH03] como un apéndice.

4.2. Entornos de desarrollo y aplicación de CoLogNetWS

El lenguaje de programación utilizado ha sido fundamentalmente Ciao Prolog, aprovechando las facilidades que ofrece, no sólo en lo referente a lo que se espera de un sistema LP/CLP sino también, y esto es lo que en este caso hace que sobresalga de entre los demás sistemas LP/CLP, de programación WWW y persistencia de datos.

Respecto a la programación WWW, como vimos en la introducción de este documento, estas ventajas competitivas radican en la librería *PiLLoW* gracias a la cual se ha hecho todo el manejo de formularios, generación dinámica de páginas Web, y manejo de ficheros XML.

Además, *PiLLoW* ha permitido mediante la creación de scripts CGI la implementación de todo el servidor, y la parte relacionada con el protocolo de intercambio de información. Por otro lado, la librería *persdb* ha facilitado la implementación de la base de datos, así como de otros predicados relacionados con el control de accesos.

CoLogNetWS puede aprovecharse de las facilidades de Ciao Prolog para generar código para las distintas plataformas hardware. Por tanto, se puede afirmar que es una aplicación multiplataforma.

4.3. Esquema de la arquitectura de la aplicación

CoLogNetWS se ha desarrollado siguiendo la arquitectura representada en la Figura 4.1. A primera vista consiste en una representación del modelo cliente-servidor, en la que por un lado, los clientes (navegador WWW) solicitan los servicios ofrecidos por la aplicación a través del protocolo HTTP y por otro lado se realiza el intercambio de datos entre CoLogNetWS y el resto de sitios Web, también a través del protocolo HTTP.

Además, se puede apreciar la existencia de cuatro módulos conceptuales sobre

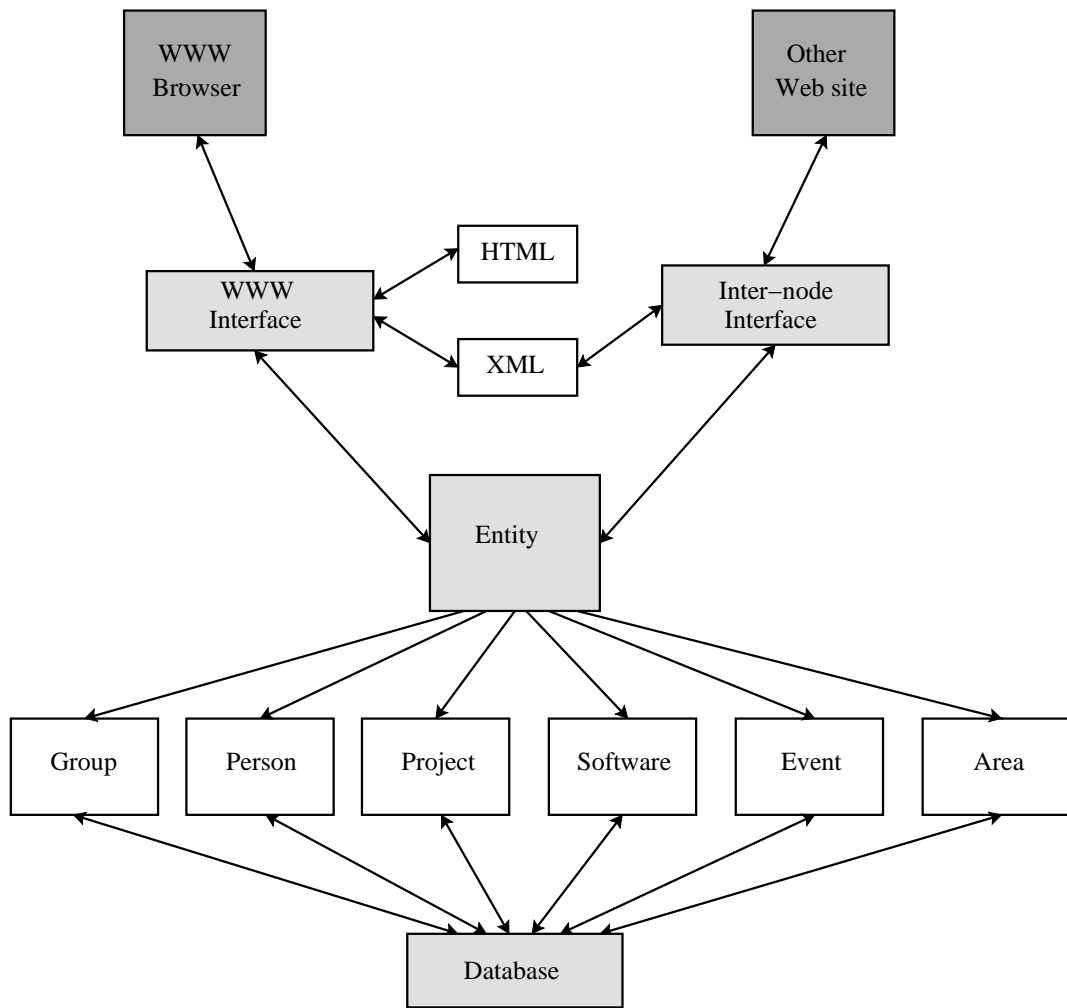


Figura 4.1: Esquema general de la arquitectura de CoLogNetWS

los que la aplicación se apoya. Estos cuatro módulos son el módulo de **Interfaz WWW**, el módulo de **Interfaz inter-nodos**, el módulo de **Entidad** y el módulo de **Base de datos**. A continuación se describen estos módulos.

4.3.1. Interfaz WWW

Este módulo es el encargado de recibir y responder a las peticiones realizadas por los usuarios Web. Básicamente interpreta los formularios HTML enviados por los usuarios, los procesa y envía la respuesta mediante la generación dinámica de páginas HTML. Consta de dos submódulos:

HTML Realiza la interpretación de los distintos formularios HTML, así como la generación dinámica de páginas HTML. Veamos cada una de estas tareas:

- Interpretación de formularios HTML. Consiste en recoger los valores de los distintos campos de los formularios HTML y operar en función de estos valores, a través del módulo “Entidad”.
- Generación dinámica de páginas HTML. Permite cumplir el objetivo descrito en el capítulo 2 de ofrecer un GUI intuitivo y fácilmente configurable. Dispone de un conjunto de plantillas HTML cuyo contenido representa un aspecto concreto del entorno gráfico del sistema.

Cada plantilla tiene una determinada estructura en función del aspecto y el tipo de información que se quiera mostrar. Todas son procesadas y convertidas en términos Prolog. Un ejemplo de una de estas páginas podría ser:

```
<html>
  <body>
    <table>
```



```

<tr><td> <V>title</V> </td></tr>
<tr><td><table> <V>entity</V> </table></td></tr>
</table>
</body>
</html>

```

donde `title` y `entity` representan trozos de código HTML, que a su vez pueden contener variables. Unos ejemplos concretos para las variables anteriores serían

- `title`:

```

Areas

```

- `entity`:

```

<tr>
<td> <table>
<tr><td>Name Abbrev:</td>
<td> <V>namea</V> </td></tr>
<tr><td>Name Full:</td>
<td> <V>namef</V></td></tr>
<td> <V>description</V></td></tr>
</table></td></tr>

```

Nuevamente vemos como `entity` contiene código HTML y además nuevas variables. En este caso la primera página representa un formato general para mostrar todas las entidades del sistema. Al procesar esta página se sustituyen

las variables por sus respectivos valores destacando la variable `entity` que se sustituiría, en este caso, por otra plantilla que muestra la información de las áreas del sistema.

Esta plantilla al ser nuevamente procesada sustituiría, en este caso, sus variables `namea`, `namef` y `description` por la información de un área en concreto. Al procesar cada página HTML, se crea una lista de términos Prolog equivalentes donde cada variable se sustituye por un hueco que contiene al parámetro y que, al ser llamado genera un trozo de código `PiLLoW` equivalente.

XML Permite ofrecer una interfaz XML, a través de la que un usuario Web u otro sitio Web pueden comunicarse con CoLogNetWS. Para ello, lleva a cabo dos tareas: generación de código XML para su posterior envío a otros sitios Web, e interpretación de código XML que permite recibir peticiones tanto de un usuario Web, como de otro sitio Web. Como se puede ver en la figura 4.1 este submódulo y el submódulo “HTML” se comunican entre sí para permitir a un usuario Web operar a través de la interfaz XML. Este submódulo realiza las siguientes tareas:

- Generación de código XML

Cuenta con un conjunto de ficheros XML cuyo contenido representa cada uno de los mensajes que se intercambian entre los distintos nodos, correspondientes a cada transacción realizada en la base de datos. Cada uno de estos mensajes tienen una determinada estructura en función del tipo de información que se quiera intercambiar. Todos los mensajes XML se generan de la misma manera: primero se genera el código relativo a la entidad sobre la que se realiza la transacción (`code_entities_xml/5`), y posteriormente se hace lo mismo con el código concerniente a la información de otras entidades asociadas a esa entidad (`code_relations_xml/3`). Un ejemplo de estos ficheros XML sería:

```
<<V>action</V>>
  <object>
    <V>object</V>
  </object>
  <associated_data>
    <V>associated_data</V>
  </associated_data>
</<V>action</V>>
```

donde `action`, `object` y `associated_data` representan trozos de código XML, que a su vez pueden contener variables. En concreto estas plantillas podrían tener los siguientes valores:

- `action`:

```
register
```

- `object`:

```
<area>
  <identifier> <V>id</V> </identifier>
  <namea> <V>namea</V> </namea>
  <namef> <V>namef</V> </namef>
  <description> <V>description</V> </description>
  <password> <V>password</V> </password>
</area>
```

- `associated_data`:

En este caso estaría vacío

Vemos como `object` contiene código XML y además nuevas variables. El primer fichero XML representa la estructura general de todos los mensajes XML que se intercambian entre los distintos nodos. Al procesar este fichero se sustituyen las variables por sus respectivos valores, y en este caso la información que se intercambiaría sería el alta de una área.

- Interpretación de código XML

Permite parsear un fichero XML y validarlo de acuerdo a un esquema XML definido previamente, documentado en el manual de la aplicación [NCH03]. Además permite procesar la información obtenida, para finalmente actualizar la base de datos de CoLogNetWS. Este proceso se realiza a través del predicado `process_file/3`, también documentado en el manual de la aplicación.

4.3.2. Interfaz inter-nodos

Permite intercambiar mensajes XML, a través del protocolo HTTP, entre los distintos nodos con el fin de tener actualizadas entre sí las bases de datos de todos los nodos. La secuencia de pasos para llevar a cabo ese intercambio consta de:

- Un nodo genera una petición HTTP cuyo cuerpo está compuesto por un mensaje XML generado previamente. Este nodo envía la petición HTTP al resto de nodos con los que se quiere comunicar.
- Cada uno de los nodos recibe la petición HTTP y procesa el contenido del mensaje XML. Una vez realizado el procesamiento, enviarán una respuesta

HTTP con el resultado de esta operación.

- El nodo que originó el intercambio recibe la respuesta HTTP, que le permite conocer si el envío se ha realizado de forma correcta o no.

Nótese que en el caso de CoLogNetWS la generación y el procesamiento del mensaje XML se realiza de la manera descrita en el submódulo XML.

4.3.3. Entidad

Este módulo es, en realidad, una generalización de un conjunto de módulos donde cada uno de ellos se ocupa de gestionar las distintas entidades sobre las que trata CoLogNetWS. Todos ellos permiten cumplir con el objetivo de, además de gestionar, operar a través de cualquier navegador WWW, ofreciendo una serie de operaciones descritas en el capítulo 2. Por tanto, su especialización proporciona los módulos: **Grupo**, **Persona**, **Proyecto**, **Software**, **Área** y **Evento**. Cada uno de ellos permite visualizar el contenido de sus distintos objetos almacenados en el sistema mediante el predicado `entity/2`, e insertar, modificar o eliminar un objeto a través del predicado `process_entity/4`.

4.3.4. Base de datos

Contiene la base de datos de la aplicación. Por tanto, representa el nivel de almacenamiento de CoLogNetWS. La base de datos está implementada mediante predicados persistentes basados en ficheros, a través de la librería `persdb` de Ciao. El principal beneficio de esta elección es que el sistema se hace robusto frente a interrupciones o apagones imprevistos gracias a que todos los datos que necesita para funcionar se guardan en predicados que mantienen su estado a través de las sucesivas ejecuciones. Por tanto, al reentrancarse tras la posible interrupción, la configuración

del sistema seguirá siendo exactamente la misma que al sufrir la parada.

4.4. Los servicios de CoLogNetWS

En esta sección describiremos los servicios ofrecidos por CoLogNetWS. De esta manera, haremos un recorrido por estos servicios, comentando sus peculiaridades, las estructuras y predicados relevantes utilizados, así como su disponibilidad para los distintos tipos de usuarios. Para más información sobre los servicios ofrecidos por CoLogNetWS, consultar el manual de referencia [NCH03] incluido como un apéndice de este documento.

En CoLogNetWS, los servicios han sido implementados como ejecutables CGI a los que el servidor permite acceder, desde clientes WWW o desde otros sitios Web.

4.4.1. Visualizar/Insertar/Modificar/Eliminar a través del interfaz WWW

Para que un usuario pueda elegir una de las operaciones ofrecidas desde la Web, se crea una sesión normal o privilegiada en función del tipo de usuario que sea. La manera de diferenciar un acceso de otro es mediante dos URL distintas, donde la correspondiente al usuario privilegiado está protegido por una autenticación basada en NCSA http.

A continuación se le muestra al usuario un menú donde puede elegir una de entre todas las entidades disponibles. Una vez elegida la entidad, CoLogNetWS lleva a cabo la operación de visualizar. Llegado a este punto, el sistema almacena un identificador asociado a esa sesión. Este identificador tiene información acerca del tipo de acceso, que puede ser normal o privilegiado, que se ha llevado a cabo. A grandes rasgos, las operaciones disponibles mediante este servicio son las siguientes:

Visualizar Para cada tipo de entidad se muestran todos sus objetos almacenados en el sistema mediante el predicados `entity/2`. Este predicado muestra los campos de información definidos para cada entidad según lo establecido en el capítulo 2, teniendo en cuenta que sólo mostrará aquellos que campos cuya información no sea nula. Otro tratamiento especial será que aquel campo de información que tenga más valores de un umbral establecido serán mostrado en una página HTML separada del resto de información general, mediante el predicado `process_entity/4`.

Además de visualizar el contenido de cada objeto, esta operación muestra un menú en el que permite realizar el resto de operaciones. Es decir, por un lado permite insertar un nuevo objeto, y por otro permite modificar o eliminar un objeto seleccionado previamente.

Insertar Permite dar de alta en el sistema, un nuevo objeto de una entidad determinada. El predicado `process_entity/4` se encarga tanto de mostrar el formulario HTML para la inserción del nuevo objeto, como también del procesamiento de dicha información.

Modificar Una vez seleccionado el objeto y elegida la opción de modificar, el sistema lleva a cabo un proceso de autenticación y muestra los campos almacenados en el sistema sobre ese objeto mediante un formulario HTML que permite modificar dichos campos. Posteriormente, el usuario modifica la información deseada y el sistema la actualiza.

El proceso de autenticación llevado a cabo por CoLogNetWS consta de los siguientes pasos. En primer lugar, comprueba qué tipo de usuario pide el servicio. En el caso de ser un usuario común, el sistema inicia un proceso de autenticación. Este tiene dos fases: en la primera se le pide al usuario un password, y en la segunda

se comprueba dicho password. Este password está asociado al objeto en cuestión y no al usuario. Si la comprobación es correcta se muestra la información del objeto, en caso contrario se le muestra un mensaje al usuario denegándole la operación. Una vez que el usuario se ha autenticado, se anota en el sistema que dicho usuario asociado a un identificador de sesión ha realizado correctamente la autenticación. Por el contrario, si se trata de un administrador, éste no tendrá que pasar por el proceso de autenticación. Por ello, podrá eliminar o modificar un ítem determinado sin ninguna restricción. Todas estas operaciones las realiza el predicado `process_entity/4`.

Eliminar Una vez seleccionado un objeto y elegida la opción de eliminar, el sistema borra de su base de datos la información del objeto y cualquier referencia a él. De la misma manera que en la operación de modificar, antes de eliminar se comprueba el tipo de usuario, y en caso de ser un usuario común es sometido al mismo proceso de autenticación. Estas operaciones las realiza el predicado `process_entity/4`.

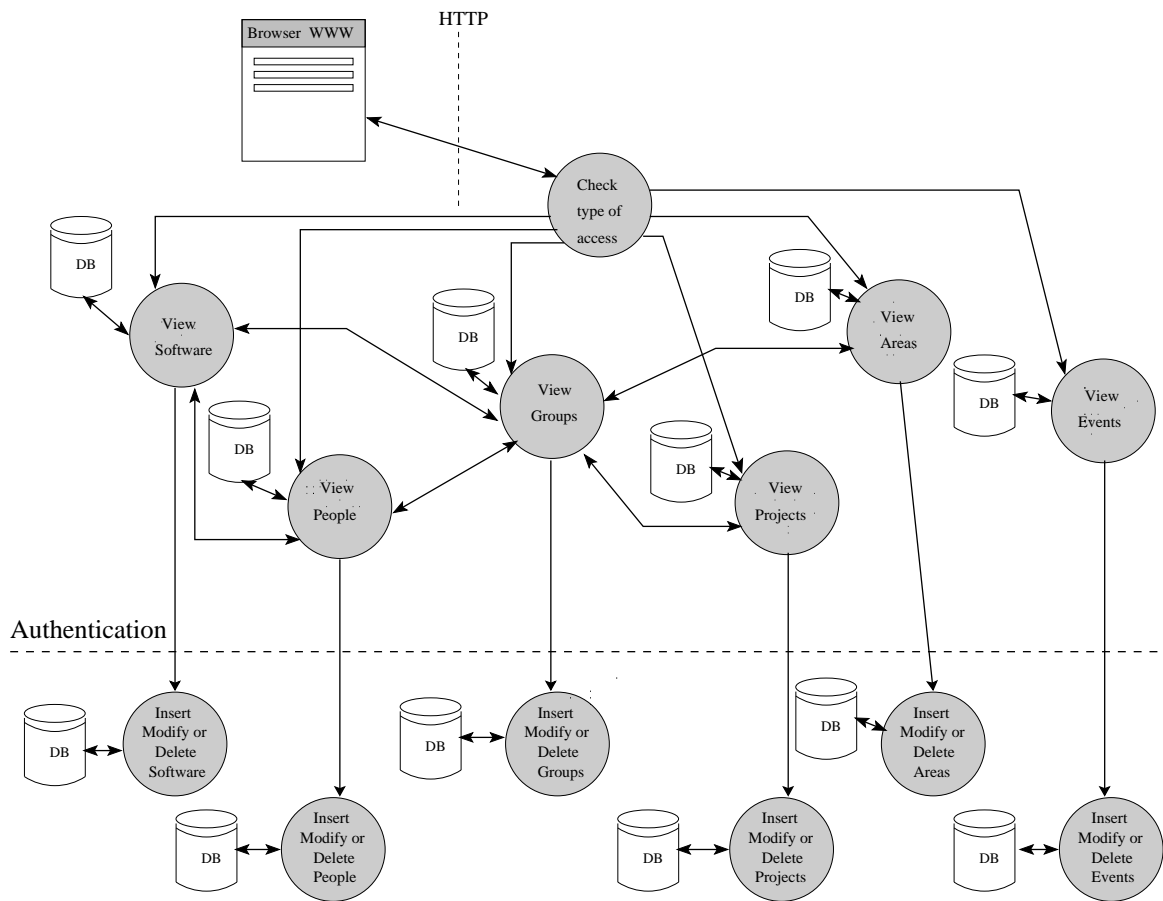


Figura 4.2: Operaciones desde WWW

4.4.2. Envío y Recepción de datos para la sincronización de las bases de datos

Este servicio permite la comunicación entre distintos sitios Web, con el objetivo de mantener actualizadas sus bases de datos. La idea general consiste en un protocolo de intercambio de datos simple que mediante el intercambio de mensajes permita realizar operaciones de manera remota. Tanto el envío como recepción de datos se basa en el protocolo definido dentro del punto 4.3.2. Las operaciones que permite este servicio son:

Envío de datos Se puede realizar de manera inmediata o modo batch. La manera de decidir el tipo de envío es mediante una variable de configuración del sistema que permite elegir una u otra modalidad. Esta variable se encuentra en el fichero de configuración `settings.pl`.

El primer tipo de envío consta de una única fase. Cada vez que se realiza una transacción en la aplicación, se envía directamente la información asociada a dicha transacción codificada en formato XML a través del predicado `data_exchange/3`. Las direcciones URL destino son consultadas a través del predicado `url_websites/2`.

La segunda modalidad consta a su vez de dos subfases:

- Recogida de los mensajes XML. Previamente la aplicación genera un fichero XML cada vez que realiza una transacción y lo deposita en unos buffers intermedios, mediante el predicado `send_buffer/2`. La fase de recogida consiste en obtener todos los ficheros XML de los buffers intermedios (`directory_files_mapped/2`). La razón por la cual hay varios buffers, es que hay uno por cada tipo de entidad. De esta manera, se hace más sencillo el envío selectivo a los distintos sitios Web en función del tipo de entidad que quiera recibir.

- Envío de los mensajes XML. Esta fase realiza el envío de los ficheros XML obtenidos en la fase anterior (`send_files/1`). Para saber a qué sitio Web se debe enviar el mensaje, se consulta el predicado `url_websites/2`, que devuelve información sobre las URLs destino.

El envío de datos se realiza a nivel interno, por lo que es ajeno al control de un usuario normal.

Recepción de datos Esta fase se realiza a través de la ejecución de un CGI cuyo argumento de entrada es un fichero XML. Durante dicha ejecución se realiza una comprobación de los datos contenidos en el fichero XML. Si esta comprobación es positiva, procesa el fichero y actualiza la base de datos (`process_file/3`). En cambio, si no es correcta se le indica al sitio Web emisor la situación, además de registrarse en el sistema, mediante el servicio de anotación y consulta de errores definido en 4.4.4.

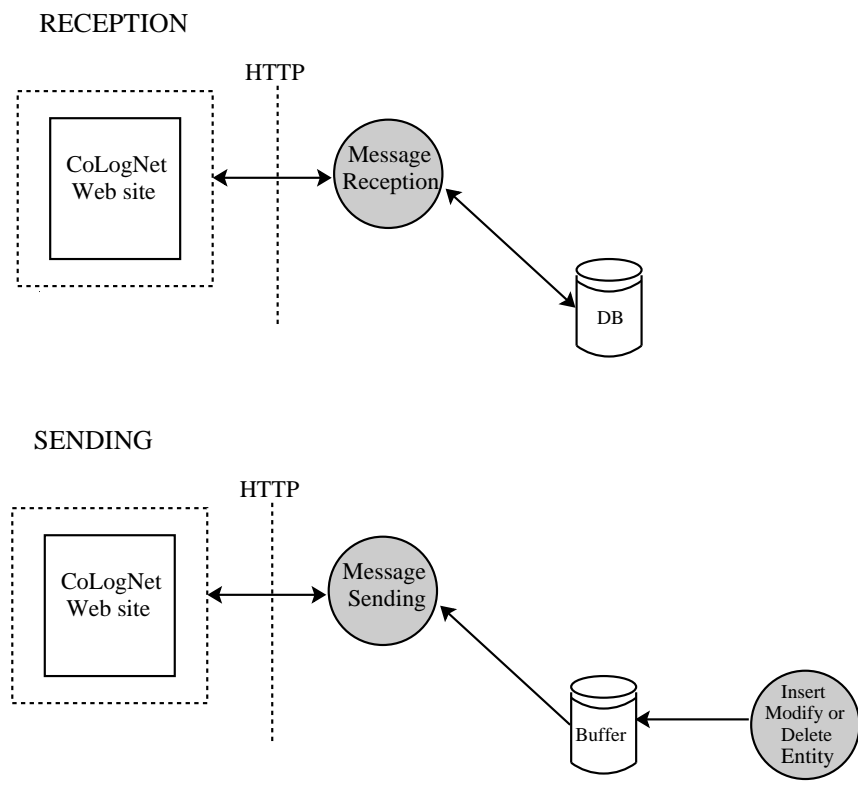


Figura 4.3: Envío y Recepción de datos a/desde otros sitios Web

4.4.3. Insertar/Modificar/Eliminar a través del interfaz XML

Este servicio permite realizar una serie de operaciones a través de un interfaz XML. Estas operaciones pueden ser remotas o desde la propia aplicación de CoLogNetWS. Está construído sobre el servicio descrito en el punto 4.4.2. Las dos posibilidades de acceder a esta interfaz XML son:

Desde el sitio Web. Permite realizar las operaciones de insertar, modificar o eliminar un objeto de una determinada entidad desde la propia aplicación. El objetivo de este servicio es facilitar al usuario dichas tareas en determinadas circunstancias. Por ejemplo, si una organización quiere registrarse en el sistema, incluyendo otra información como sus miembros, sus proyectos, o áreas de investigación puede ser más cómodo realizarlo a través de este interfaz, que desde una interfaz Web basado en menús. Esta posibilidad es accesible tanto desde una sesión de usuario normal como una sesión de administrador.

Desde otros sitios Web. Permite la recepción de datos, enviados por otros sitios Web, permitiendo la actualización de las bases de datos.

La información que se puede insertar, modificar o eliminar desde este interfaz es la misma que la descrita en el punto 4.4.1. Cabe resaltar que la modificación y el borrado de un objeto determinado obliga a proporcionar su password. Si esta password no es correcta, no se permite llevar a cabo tales operaciones.

4.4.4. Anotación / Consulta de las transacciones remotas

Este servicio permite conocer el estado final de cada uno de los intercambios de mensajes realizados con otros sitios Web. Sólo es accesible por el administrador de CoLogNetWS, siendo por tanto de ámbito privado. Este servicio permite que se

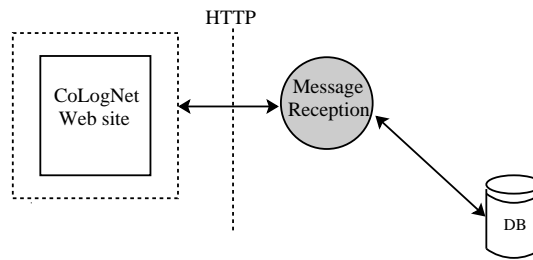


Figura 4.4: Operaciones desde otros sitios Web

pueda conocer de manera completa el estado de consistencia de la base de datos de CoLogNetWS y parcialmente el de resto de sitios Web.

Cuando se accede a la aplicación en modo privilegiado, además del menú general, existe una opción para consultar el estado de las transacciones remotas. El sistema reacciona consultando un fichero de log, que registra los errores producidos en los intercambios (sólo errores de XML), mostrándolos en una página HTML (`show_errors/4`). Para cada uno de ellos indica el tipo de error, el nombre del fichero XML, la fecha en la que se produjo el error, y el tipo de intercambio (envío o recepción). La manera de generar ese fichero de log depende de si el error se produce en un envío o en una recepción. En ambos casos el error es notificado mediante el predicado `notify_error_xml/1`:

Error XML en un envío. Una vez enviado el mensaje XML se espera la respuesta del otro sitio Web. Si la respuesta no es correcta, se escribe en el fichero de log, el tipo de error producido.

Error XML en una recepción. Si durante la validación del fichero XML se detecta un error, se le notifica al sitio Web emisor, y además se anota el error en el fichero log correspondiente.

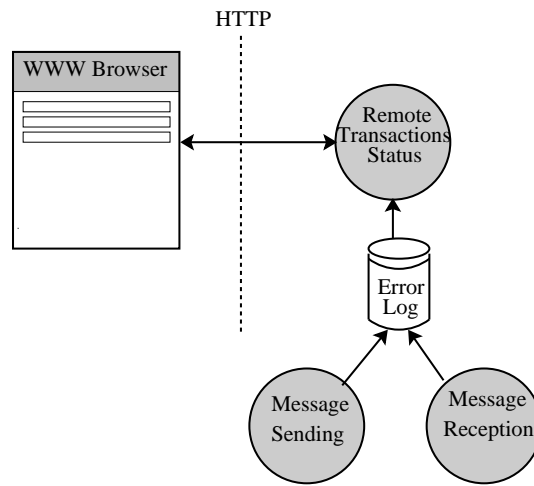


Figura 4.5: Anotación / Consulta de las transacciones remotas

Capítulo 5

Conclusiones y Trabajo futuro

5.1. Conclusiones: Consideraciones finales sobre CoLogNetWS

A través de estas conclusiones, en este punto se pretende, a modo de recapitulación, enumerar las características de CoLogNetWS que han surgido como respuesta a los objetivos marcados en la sección 2.

Las características principales de la aplicación son las siguientes:

- Entorno gráfico reconfigurable. CoLogNetWS ofrece un interfaz intuitivo y fácilmente reconfigurable. A tal fin se ha dispuesto de un sistema de plantillas en formato HTML que permiten cambiar la apariencia.
- Protocolo de comunicación con otros sitios Web. CoLogNetWS define e implementa un sencillo protocolo de intercambio de datos que permite comunicarse con otros sitios Web, a fin de tener las bases de datos actualizadas.
- Gestión de las entidades grupos, personas, proyectos, software, eventos y áreas. La aplicación permite gestionar la información definida en el capítulo 2 para cada una de las entidades. Además de la interfaz WWW disponible para poder operar con ellas, gracias a la flexibilidad del protocolo de intercambio de información, CoLogNetWS ofrece al usuario una funcionalidad adicional, construída sobre el módulo de recepción de dicho protocolo. Esta funcionalidad consiste en que el usuario normal pueda insertar, modificar o eliminar un objeto de una entidad determinada a través de un interfaz XML.

Siguiendo con la característica de flexibilidad, CoLogNetWS permite de manera sencilla el aumentar el número de entidades a gestionar, así como la información sobre cada una de ellas. Esta última característica se consigue gracias al uso del paquete `argnames` el sistema CIAO, entre otros. Este paquete permite definir estructuras que almacenan los diferentes campos de cada objeto de una manera flexible, permitiendo fácilmente añadir nuevos campos o quitar campos ya existentes.

- Utilización de predicados persistentes basados en ficheros para el almacenamiento de datos del sistema como, por ejemplo, la información relativa a un grupo o a cualquier otra entidad.

De esta manera, obtenemos varias ventajas:

- Podemos utilizar las primitivas ofrecidas por la librería `persdb` para introducir o eliminar información en la base de datos.
- Los hechos pertenecientes a predicados persistentes basados en ficheros se almacenan en formato ASCII, sin ningún tipo de formato añadido. Esto permite, en caso de que se considere necesario, editar estos ficheros y cambiar, modificar o incluir nuevos hechos, a diferencia de lo que ocurre en otros sistemas de bases de datos, en los que sólo se puede acceder a los datos del sistema a través del interfaz estándar.

La ventaja principal, de la que CoLogNetWS se aprovecha es que, si el sistema estuviera inhabilitado por cualquier razón, estos datos aún serían accesibles mediante un sencillo editor de texto.

- Modos de acceso distintos para el administrador del sistema y los usuarios comunes. De esta forma se consigue acotar la disponibilidad de los servicios de CoLogNetWS, asegurando que son utilizados por el usuario correcto.

- Obtención de una buena respuesta del sistema al satisfacer cualquier transacción, siempre y cuando la velocidad y la capacidad de la máquina que lo albergue sea razonable. CoLogNetWS se basa en una arquitectura cliente-servidor implementada sobre un conjunto de ejecutables CGI.

5.2. Trabajo futuro

En la línea de desarrollo de este proyecto se podría intentar avanzar en ciertos aspectos. Estos, bajo nuestro punto de vista, son los siguientes:

- Ampliar el protocolo de comunicación, mediante nuevos mensajes. Así por ejemplo, se podría crear una única base de datos centralizada y que todos los sitios web accedieran a esos datos mediante intercambio de nuevos mensajes creados para ese fin.
- Ampliar los servicios ofrecidos, tanto para usuarios normales como administradores, para poder cubrir más y mejor las necesidades de éstos. Por ejemplo, crear una entidad “Publicaciones” que permita conocer todo el material publicado por un investigador o por un grupo.
- Dar mayor seguridad, tanto a CoLogNetWs como al protocolo de comunicación. Por ejemplo, conseguir confidencialidad e integridad en el intercambio de datos, o en el paso de datos de una página HTML a otra, mediante protocolos de comunicaciones seguros como HTTPS.
- Ampliar la capacidad de la base de datos. En este sentido habría que incluir los conceptos de clave primaria, clave extranjera e integridad referencial, es decir, acercar más el sistema a las especificaciones de una base de datos relacional. Se podría conseguir a través de un interfaz de base de datos Prolog-base de

datos relacionales, dotando al lenguaje Ciao-Prolog de una capa semántica que garantice dicha integridad.

- Utilizar el lenguaje de consulta definido en el proyecto eBroker. De esta manera se conseguiría mayor flexibilidad a la hora de tratar un fichero XML.

Bibliografía

- [BLFIF96] T. Berners-Lee, R. Fielding, UC. Irvine y H. Frystyc. Hypertext Transfer Protocol http/1.0, Mayo 1996. Available from <http://www.w3.org/Protocols/rfc1945/rfc1945>.
- [CCG⁺98] I. Caballero, D. Cabeza, S. Genaim, J.M. Gomez y M. Hermenegildo. persdb_sql: SQL Persistent Database Interface. Technical Report D3.1.M2-A2 CLIP10/98.0, RADIOWEB Project, Diciembre 1998.
- [CHGT98] D. Cabeza, M. Hermenegildo, S. Genaim y C. Taboch. Layout and style (last) language, a preliminary design. Technical Report D2.2.M2, RADIOWEB Project, Diciembre 1998.
- [CHV99] D. Cabeza, M. Hermenegildo y S. Varma. The PiLLoW/Ciao Library for INTERNET/WWW Programming using Computational Logic Systems, Mayo 1999. See <http://www.clip.dia.fi.upm.es/Software/pillow/pillow.html>.
- [GCH98a] J.M. Gomez, D. Cabeza y M. Hermenegildo. persdb: Persistent Database Interface. Technical Report D3.1.M2-A1 CLIP9/98.0, RADIOWEB Project, Diciembre 1998.
- [GCH98b] J.M. Gomez, D. Cabeza y M. Hermenegildo. WebDB: A Database WWW Interface. Technical Report D3.1.M2-A3 CLIP11/98.0, RADIOWEB Project, Diciembre 1998.

[NCH03] J.A. Navas, D. Cabeza y M. Hermenegildo. CoLogNetWS: Development of a Web Site with Database Multinode Distributed Update. Technical Report CLIP6/2003.0, CoLogNet Project, Noviembre 2003.

Apéndice A

Manual de referencia de CoLogNetWS