

DISTRIBUCIÓN DE APLICACIONES VIA SISTEMA CLIENTE-SERVIDOR

Presentación del proyecto ATF

El sistema telemático que aquí se presenta (ATF o Asistente Telemático a la Formación) es un desarrollo de Divisa Informática S.A., empresa de Ingeniería en Informática y Telecomunicaciones ubicada en Castilla y León, en el que colabora la Universidad de Valladolid a través de la Escuela Técnica Superior de Ingenieros de Telecomunicación de Valladolid.

Para encontrar una descripción detallada, dirigirse a la página oficial del proyecto: <http://atf.dvnet.es>.

Arquitectura del sistema ATF. Servidor Educativo

La arquitectura definida, sigue la estructura tradicional cliente servidor (Fig. 1), aunque esta se encuentra modificada de forma parcial:

donde el Servidor Educativo es el compendio de todos los servidores asociados, localizados en el proveedor de servicios telemáticos. Un requisito necesario a la hora de diseñar la arquitectura, fue el considerar una distribución de estos en diferentes máquinas, tal que se permita su futura escalabilidad. Además la arquitectura elegida deberá ser fácilmente repetible. En sí, no se ofrece una arquitectura específica, sino una filosofía de arquitectura, lo suficientemente abierta como para no dirigirse a un sector específico (una institución puede reutilizar sus recursos disponibles o compartirlos con otro organismo para implementar el sistema o parte de él)

Una relación de servidores involucrados bajo ATF y que componen el Servidor Distribuido Educativo puede ser esta:

Servidor Web Base de Datos Relacional Servidor SMTP Servidor POP3 Servidor News Servidor FTP anónimo

La función del servidor Web, es actuar a modo de Servidor de Aplicación. El cliente, accede a través del servidor Web a los contenidos del curso y otros módulos adicionales bajo descarga selectiva. También actúa de interfaz de acceso a la base de datos aunque el cliente pueda consultar directamente la base de datos con el fin de obtener cierta información administrativa (por ejemplo para una autoconfiguración selectiva de la herramienta del cliente)

El cliente del sistema ATF

El cliente, denominado Aula Virtual y desarrollado para sistemas operativos Windows95 y NT, realiza las funciones fundamentales de un navegador tradicional, siendo la interfaz personal de trabajo del alumno y

profesor.

Funcionalmente, se descompone en dos áreas. Una sobre la cual se localizan las herramientas del cliente (la parte izquierda de la Fig. 2). El otro área, permite navegar y visualizar los contenidos de los cursos. (la parte derecha de la Fig. 2)

Como aspecto favorable a la interfaz, el alumno nunca asocia los contenidos a direcciones de formato URL (o cualquier otro formato no entendible). De una forma más pedagógica, los contenidos se pueden relacionar con frases breves (herramienta de favoritos) o a textos personales más largos (herramienta de notas) creados por el mismo alumno.

La interfaz también dispone de una herramienta de correo. Otra herramienta interesante desarrollada fue el cliente de news, aquí bajo la denominación de tablón de anuncios. Estos dos clientes permiten la comunicación directa entre los participantes del curso y su tutor o tutores. Al implementar estas herramientas, se buscó la facilidad de uso. No son herramientas avanzadas o complejas, por contra, son altamente intuitivas en su manejo. El usuario muchas veces utiliza un porcentaje muy limitado de la funcionalidad de su correo, ya que el esfuerzo asociado al aprendizaje siempre es alto. Un objetivo que guía el desarrollo es no desmotivar nunca al usuario, obligándole aprender a utilizar un entorno complejo, más aún si no se encuentra familiarizado con las redes telemáticas. También, como se advierte en la figura, el atractivo visual se consideró como un elemento importante, que incita al usuario, y lo introduce en el entorno ficticio del aula.

La interfaz de cliente incorpora otras herramientas importantes: Aquellas asociadas al acceso al curso y su gestión.

Junto a estas, se dispone de la posibilidad de realizar los exámenes propuestos por el tutor o acceder a los eventos definidos por este (por ejemplo, avisos de clases presenciales). Los eventos se definen como aquellos hechos o noticias de especial relevancia que emitidos por el profesor, deben ocupar un papel diferenciado del resto de noticias del tablón.

Sin embargo, el diseño de ATF no busca que su interfaz cliente se restrinja al aula virtual desarrollado. No se pensó como un desarrollo cerrado. La interfaz puede estar constituida por cualquier navegador comercial. De esta forma, los servicios ofrecidos por el sistema, pueden ser disfrutados por usuarios (al menos en su mayor parte) que no dispongan del cliente del Aula Virtual. El Aula Virtual, no obstante, ofrece una normalización de acceso al sistema y a los cursos educativos, junto con el uso de las funcionalidades ofrecidas del sistema ATF.

PROCESAMIENTO PARALELO DE APLICACIONES

LIBRERIA PPL

(PARALLEL PROGRAMMING LIBRARY)

PROTOCOLOS DE APLICACION STANDARD Y NO STANDARD

Cualquier programador que haga uso de la librería de funciones para la implementación de una aplicación que realice su labor en paralelo tendrá que crear dos programas o procesos distintos :

El servidor, encargado de implementar la función de proceso sobre un conjunto o zona de datos con una estructura concreta.

El cliente, encargado de la interacción con el usuario y de controlar la operación de los servidores.

Para coordinar la ejecución de los procesos cliente y servidor es necesario establecer un protocolo de comunicación entre ambos, es decir un serie de normas que han de ser conocidas por los dos procesos y que han de ser cumplimentadas para que una solicitud de servicio pueda llevarse a cabo de forma correcta. Esto es lo que se conoce como Protocolo de Aplicación.

El protocolo TCP/IP incluye muchos protocolos de aplicación, y diariamente aparecen nuevos protocolos de aplicación. De hecho, cada vez que un programador desarrolla un programa distribuido que emplea el TCP/IP para comunicar los procesos participantes, diseña un nuevo protocolo de aplicación.

Existe un conjunto de protocolos de aplicación que han sido documentados en RFC y adoptados como parte oficial del protocolo TCP/IP. A tales protocolos se les denomina protocolos de aplicación standard. Otros protocolos, ideados por los programadores de aplicaciones para su uso privado, se les conoce como protocolos de aplicación no standard.

Ejemplos de protocolos de aplicación standard son el FTP (File Transfer Protocol), login remoto (TELNET) y correo electrónico (E-MAIL). La sesión remota (remote login) es una de las aplicaciones TCP/IP más populares.

ESQUEMA DE FUNCIONAMIENTO DEL SERVIDOR

El proceso servidor es el encargado de llevar a cabo el procesamiento de los datos remitidos por el proceso cliente. Su función es quedar en espera de peticiones de servicio y atenderlas conforme le van llegando.

Cuando el proceso servidor es ejecutado en uno de los hosts o equipos de la red, realiza la siguientes tareas de inicialización :

Averigua la dirección IP, así como el nombre (si es que hay alguno definido) del host (anfitrión) en el cual se está ejecutando. Realiza una llamada al sistema para obtener el descriptor de socket a través de cual recibirá y transmitirá las tramas de bytes. Forma la dirección de red en la cual va a escuchar peticiones de servicio. Dicha dirección está constituida por la dirección IP del host y el puerto en el cual va a atender solicitudes de servicio. Se bloquea en espera de peticiones de servicio.

SERVICIOS PPL STANDARD

Los procesos cliente y servidor definidos en la librería PPL utilizan un protocolo de aplicación, lo llamaremos protocolo PPL, el cual define los servicios más comunes que van a ser empleados por las aplicaciones paralelas que van a poder ser desarrolladas con PPL. A estos servicios predefinidos se los denominaremos Servicios PPL Standard. El programador que haga uso de la librería PPL es libre de añadir cuantos servicios adicionales sean necesarios para la aplicación que esté desarrollando, del mismo modo puede modificar los servicios predefinidos para adaptarlos a sus necesidades. Veamos detalladamente cada uno de los servicios predefinidos por el protocolo PPL:

SERVICIO PING

No se puede definir como un auténtico servicio, ya que el servidor no lleva a cabo ninguna labor de proceso o transferencia de datos. Es empleado por el cliente cuando se encuentra examinando la red en busca de servidores para saber si en la dirección IP que está siendo examinada se encuentra algún proceso servidor activo.

SERVICIO INICIO_PROCESO

Este servicio indica al servidor que aplique la función de proceso sobre el conjunto de datos o partición que previamente tiene que haberle transmitido el cliente.

El cliente suele mantener una lista con las direcciones de red de los servidores activos. Una vez enviada la partición a cada uno de los servidores se puede emplear la función de librería `start_proces(struct t_host* lista_servers)` la cual se encarga de iniciar el proceso de ejecución en cada uno de los servidores de la lista que recibe como parámetro.

Protocolo INICIO_PROCESO.

El cliente compone y manda la siguiente trama al servidor para indicarle el servicio de proceso de datos :

Recibida la trama de petición de servicio, el servidor genera un proceso hijo que se encarga de llevar a cabo la labor de proceso. Concluido el proceso de datos es el propio hijo el encargado de componer la siguiente trama de respuesta que indica al cliente la conclusión del proceso de la zona de datos y, por lo tanto, la disponibilidad para transmitir los resultados.

Una vez transmitida la trama de solicitud de proceso de datos, el cliente queda en espera de la trama que indica el fin del proceso. Es conveniente que el cliente establezca un timeout con el tiempo de proceso estimado para evitar que espere eternamente si, por ejemplo, el servidor cae a mitad de proceso o merma considerablemente su velocidad de proceso.

SERVICIO GET_DATA

Este servicio es empleado por el cliente para obtener los resultado del procesamiento llevado a cabo por el servidor sobre una zona de datos previamente transmitida por el cliente al servidor.

Protocolo GET_DATA.

El cliente compone una trama en cuya cabecera se indica al servidor que desea el servicio GET_DATA.

Si no hay nada que lo impida, el servidor responde al cliente con la siguiente trama :

en la cual le indica que se va a iniciar la transmisión de bytes, así como la cantidad de bytes a transferir.

Acto seguido pasa a transmitir los bytes solicitados al cliente. Para llevar a cabo la transferencia se sirve de la función `PPL envia_bytes(int sfd, char* buf, int nbytes)`, la cual se encarga de enviar, a través del socket `sfd` la cantidad de bytes `nbytes` apuntada por el puntero `buf`. Por su parte el cliente se sirve de la función `recibe_bytes(int sfd, char* buf, int nbytes)`, para recibir los bytes solicitados.

Concluida la transferencia el servidor se bloquea en espera de nuevas solicitudes de servicio.

Esquema Gráfico GET_DATA.

SERVICIO STATUS

El cliente solicita este servicio del servidor cuando desea averiguar el estado en el cual se encuentra el servidor (procesando, libre,)

Protocolo STATUS.

El proceso cliente compone una trama con la siguiente estructura :

A continuación pasa a establecer conexión con el servidor (connect_server).

Establecida la conexión, el cliente pasa a transmitir al servidor la trama anteriormente expuesta. El servidor analiza el servicio solicitado, STATUS en este caso, y compone una trama de respuesta con el siguiente formato :

El servidor almacena su estado en una variable almacenada en memoria compartida (para que pueda ser modificada por procesos hijo). Este servicio lo único que hace es retornar el valor de dicha variable al cliente que inicio la solicitud.

Este servicio puede ser de utilidad para, concluido un tiempo de espera, solicitar el estado del servidor para saber si aún está operativo o ha dejado de funcionar.

Esquema Gráfico STATUS.

CONTROL DE PROCESAMIENTO CONCURRENTES

Como parte del proyecto Modelado y Simulación Neuronal en el Web, se ha completado la versión 3.0 del sistema de simulación para redes neuronales de gran escala NSL descrito en libro el Lenguaje de Simulación Neuronal (NSL por sus siglas en inglés). NSL provee un ambiente de simulación que simplifica la tarea de modelar redes neuronales, apoyando, en particular, modelos neuronales que tienen como estructura de datos capas de neuronas sencillas con propiedades y patrones de conexión similares, donde las neuronas se modelan como integradores con fuga y las interconexiones están sujetas a diferentes reglas de aprendizaje. NSL sigue un diseño orientado a objetos, ofreciendo abstracciones de programación de alto nivel correspondientes a elementos neuronales. Actualmente se está desarrollando en NSL la liga entre múltiples niveles de modelado neuronal pudiendo expresar el hecho de que las propias redes neuronales pueden ser interconectadas de manera jerárquica, creando ensamblajes como medio de composición, tales como los que se describen en el Lenguaje de Esquemas Abstracto (ASL por sus siglas en inglés). ASL, lenguaje integrado al de NSL, incorpora además aspectos de la programación orientada a objetos concurrentes, donde un esquema es representado como un template del cual muchas instancias pueden ser creadas. ASL utiliza un modelo jerárquico, permitiendo diseños de arriba hacia abajo y de abajo hacia arriba, apoyado por un lenguaje concurrente que permite una implementación distribuida, además de integrarse con las redes neuronales descritas en NSL. Otras características de ASL son su naturaleza dinámica y asíncrona, y la inclusión de ensamblajes de esquemas dinámicos como base para la composición. De acuerdo a su comportamiento, se describe cómo las instancias de un esquema responderán a comunicaciones externas. Cada instancia de un esquema cuenta con un conjunto de múltiples puertos de entrada y salida a través de los cuales la comunicación toma lugar. Un ensamblaje de esquemas, la base para agregación, es una red de instancias de esquemas, y puede considerarse un esquema para su procesamiento. Cómo un esquema puede ser descompuesto en cualquier número de "sub-esquemas", pudieran haber virtualmente cualquier número de niveles de abstracción.

El sistema integrado NSL/ASL tiene como objetivo apoyar la simulación en los niveles de agentes robóticos autónomos, comportamiento de agentes y redes neuronales, incorporando los siguientes componentes: lenguajes compilados para describir modelos, lenguajes interpretados para controlar interactivamente las simulaciones, lenguajes de programación visual, interfaces gráficas, técnicas de visualización, procesamiento concurrente (múltiples hilos) y distribuido, arquitecturas meta-nivel, herramientas de análisis numéricos y metodologías de simulación. Existen dos implementaciones de NSL, NSLJ en Java y NSLC en C++. El repositorio de modelos para NSL/ASL tiene como objetivo guardar cuatro tipos de estructuras básicas en la base de datos con acceso directo desde el Web:

Modelos: Vistas de alto nivel de un modelo, ligado a los elementos descritos a continuación. **Módulos:** Componentes de un modelo estructurados jerárquicamente. **Simulaciones:** Para cada ejecución "útil" de un modelo, se guarda los parámetros y valores de entrada, además de anotarse aspectos particulares en relación a los resultados. **Interfaces:** Para apoyo en el control y visualización, se incluyen interfaces especializadas que permiten interactuar de manera diferente con cada modelo y simulaciones correspondientes.

El sistema de simulación NSL versión 3.0 (C++ y Java) junto con la biblioteca de modelos (algunos de los modelos son: Backpropagation, Hopfield, Seleccionador Máximo, Retina, Control Ocular, Cerebelo, Aprendizaje Condicional, Aprendizaje con Refuerzo, Aprendizaje de Evitar Obstáculo, Reconocimiento de Caras) ya se encuentran en el Web (<http://cannes.rhon.itam.mx>). Actualmente se está completando el manejo de configuraciones para apoyo a distintos ambientes de procesamiento de manera automática incluyendo MS Windows, Unix y Linux.

ACTIVADORES Y RECUPERACION

Búsqueda acceso y recuperación de información sanitaria en bases de datos a través de www

Se plantea un nuevo método para la búsqueda, acceso y recuperación de información dispersa en bases de datos sanitarias heterogéneas a través de la World Wide Web. Como primer logro, se ha creado un prototipo que permite el acceso a bases de datos

sanitarias con un modelo predefinido. Se está trabajando en un nuevo prototipo, basado en una arquitectura distribuida, que permita el acceso a distintos modelos de bases de datos creando una visión unificada y virtual de todas ellas. Los distintos prototipos desarrollados serán evaluados entre los investigadores del Grupo de Informática Médica (FI/U.PM) y miembros del Instituto de Salud Carlos III.

Un factor común que subyace a los diferentes problemas del sistema sanitario español (listas de espera, elección libre de médico, difícil acceso a los mejores medios diagnósticos y terapéuticos, desconocimiento de los procedimientos utilizados en cada unidad sanitaria) es que el manejo de la información sanitaria es clave para su eficaz funcionamiento. La creación de un método que permitiese modelizar información sanitaria de los distintos centros y su recuperación inmediata desde lugares remotos ayudaría a la resolución de estos problemas.

La recogida de información sobre las actividades de un centro sanitario se realiza, generalmente, a través de cuestionarios. Estos son enviados a los distintos centros y rellenados por el personal de los mismos. Es un proceso lento y no siempre los datos obtenidos son actuales, por lo que sería aconsejable el uso de sistemas y redes informáticas que permitiesen el acceso a esta información y su análisis en tiempo real.

En los últimos años, la World Wide Web (WWW) se ha consolidado como medio para el intercambio electrónico de datos, información y conocimientos. Son cada vez más numerosas las organizaciones que utilizan Internet para llevar a cabo sus proyectos comunes de colaboración. Independientemente del tipo de ordenador que el usuario utilice.

la información puede ser accedida y recuperada de un modo inmediato. Gracias a la WWW sería posible el diseño de nuevos métodos de acceso a la información sanitaria localizada en los distintos centros del Sistema Nacional de Salud, para su utilización efectiva.

El proyecto ARMEDA, presentado en este artículo, propone un método para la búsqueda, acceso y recuperación de información procedente de bases de datos sanitarias básicas a través de la WWW.

Dado que no existe ningún modelo estándar para bases de datos sanitarias, se creó un modelo lógico propio que recogiera información sanitaria básica. Este modelo podría ser fácilmente modificado para su uso efectivo en cualquier unidad o servicio del Sistema Nacional de Salud.

MÉTODO

Como primer logro del proyecto ARME-DA, se ha desarrollado un prototipo. En el desarrollo de este prototipo se ha seguido el estándar ESA (PSS-05)[1] para la obtención de requisitos de usuario [2], y el modelo de objetos funcional del método OMT, Object Modelling Technique [3] para el diseño.

Una vez concluido este primer prototipo, se está trabajando en dos nuevas líneas. La primera consiste en una arquitectura cliente-servidor, en la que se sustituye la CGI, Common Gateway Interface, por una applet Java. En el primer prototipo, la CCI recibía las peticiones del cliente Web y accedía a las bases de datos devolviendo al cliente el resultado; en la nueva versión, este proceso será ejecutado en el cliente web desde una applet Java. De esta forma, los accesos a las bases de datos remotas se harán directamente desde el cliente Web sin tener que pasar por una CGI, ni intercambiar páginas Web entre cliente y servidor Web.

La segunda línea de trabajo consiste en el desarrollo de una arquitectura distribuida basada en una red de agentes software. Esta arquitectura permitirá tener los procesos de búsqueda, selección y recuperación de información dispersos en distintos componentes especializados en la resolución de las tareas que tienen que realizar. Esta arquitectura incluirá el acceso a bases de datos sanitarias que sigan modelos diferentes.