

Propósito del prototipo:

Mi propuesta consiste en crear un portal Web que contemple las siguientes funcionalidades:

1º. Mostrar noticias y eventos propios del grupo de personas que administren la Web.

2º. Tener una zona de descargas a la que puedan acceder de forma sencilla y amigable los visitantes de la Web.

3º. Que los administradores puedan subir sus archivos a la zona de descarga de la forma más sencilla posible. Habitualmente, cuando se añade una descarga a la Web (la descarga puede hacerse desde otro sitio o directamente poniendo nuestro archivo en el servidor) es necesario realizar dos operaciones: En primer lugar subir el archivo al servidor y después poner una breve descripción del mismo en el portal, para que el usuario pueda identificarla. Para solucionar este problema existen dos posibilidades: una on line, que consiste en subir el fichero a Internet desde una pagina Web con un formulario, lo que requiere estar conectado para poder realizar las operaciones. Otra opción es off line, creando un archivo de descripciones unido al archivo de la descarga de forma que cuando se dispusiera de una conexión a Internet, a través de un protocolo muy conocido (FTP) podría subirlo a un sitio previamente definido permitiendo que fuera la aplicación la encargada de gestionarlo.

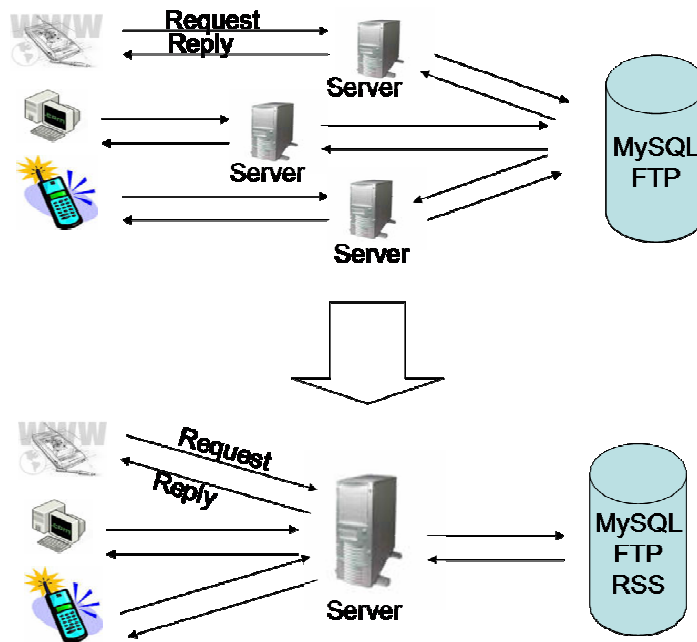
4º. Mostrar noticias procedentes de otras fuentes de Internet de forma que aparezcan en su propia Web de usuario en el momento en que las fuentes originales se publiquen sin necesidad de editarla y colgarla. Un problema importante, al que se enfrenta cualquier portal en Internet es el de mostrar información on line debido al corto periodo de tiempo en que esta información puede permanecer de actualidad. Un ejemplo son las noticias de los periódicos digitales, que pueden estar ya anticuadas al poco tiempo de generarse, lo que obliga a tener que estar pendiente de los nuevos acontecimientos y de publicarlos en la Web de la forma más rápida posible, a costa de, en algunas ocasiones, perder o malinterpretar parte de la información.

A este problema se le añaden otros. Por un lado, cuando un usuario se plantea buscar una información, siempre acudirá antes a una Web especializada que a un portal de diferentes contenidos. Por otra parte si un usuario lee varios periódicos (ya sea de distintas zonas de su país, o de distintos países) se ve obligado a abrir un gran número de ventanas de su explorador lo que complica considerablemente la forma de ver las noticias. Esto adquiriría especial dificultad para los usuarios que estén conectándose desde un dispositivo móvil.

La propuesta que hacemos solucionaría estos problemas. El administrador Web podría adquirir la información procedente de Internet de forma directa y adaptarla de forma automática y sin coste alguno a los contenidos de su portal y por parte del usuario podría seleccionar que tipo de noticias quiere ver y de que fuentes con solo conectarse a nuestra aplicación y en una sola ventana.

5°. **Que la aplicación sea funcional para cualquier dispositivo, teléfono móvil, Pda u ordenador.** Actualmente muchas aplicaciones Web están diseñadas de tal forma que si se accede desde un teléfono móvil, PDA u ordenador, se accede a una aplicación diseñada exclusivamente para funcionar solo con un dispositivo y no con los demás. Esto obliga a tener una aplicación diferente para cada dispositivo y si además todas las aplicaciones acceden a la misma base de datos, se complica enormemente el diseño.

Este esquema representa como se estructuran muchas aplicaciones y como se va a estructurar la idea:



Descripción tecnológica del prototipo

Con tecnologías como XHTML Basic, Css mobile y SVG (Tiny and Basic) podremos mostrar contenidos en dispositivos móviles. También se usará XHTML 1.1 para mostrarlos en el caso de los exploradores que sean compatibles con este tipo de estándar y por último para aquellos usuarios que utilicen habitualmente lectores de noticias en RSS usaremos RDF (RSS 1.0).

Para la comunicación entre el service Requester y el service Provider se empleara SOAP y RDF para la operación inversa.

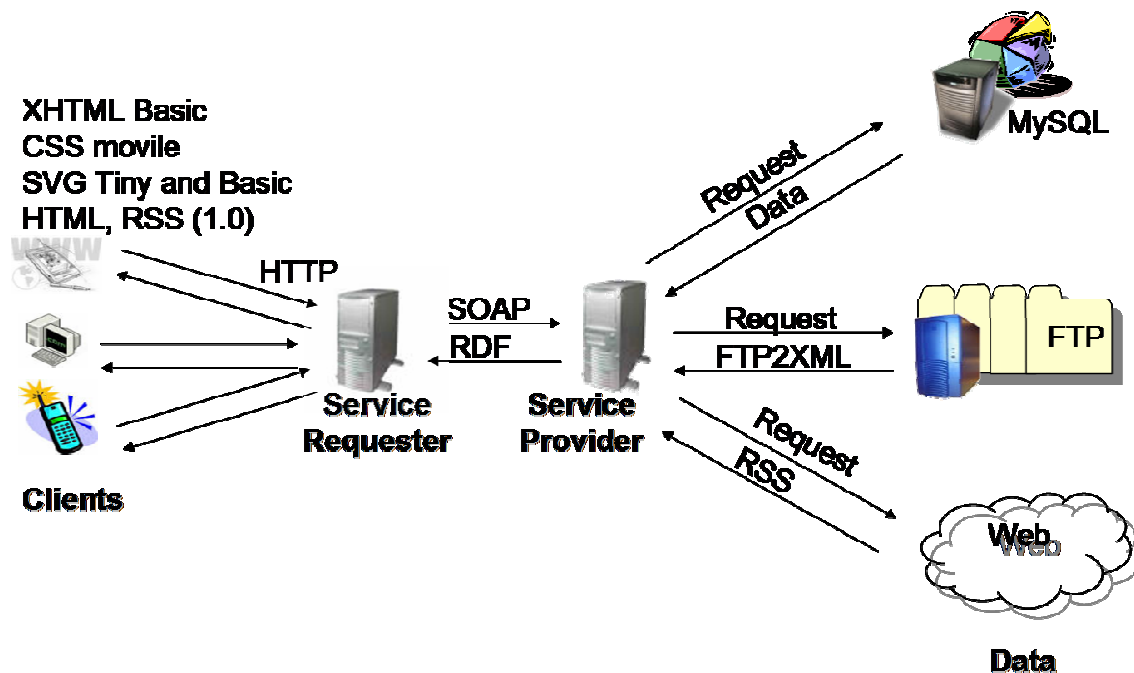
Además para la obtención de información por parte del Service provider se empleara el lenguaje XML como vía para la comunicación.

Descripción funcional

Con las nuevas tecnologías existentes (como el acceso directo a Internet desde distintos dispositivos y no solo desde un ordenador) y los nuevos estándares (como

XHTML, HTML, XHTML Basic) ya es posible identificar que tipo de explorador se encuentra conectado a nuestra aplicación y mostrarle unos u otros contenidos dependiendo de sus necesidades.

Para el desarrollo de esta aplicación podremos hacer uso de las nuevas tecnologías como RSS (que utilizaremos para obtener información en tiempo real, como las noticias de periódicos), SVG para móviles (SVG Tiny y SVG Basic) y XHTML Basic que cada vez soportan más dispositivos portátiles y que se podrían emplear como se muestra en la siguiente imagen:



El funcionamiento de esta aplicación podemos explicarla desde dos vertientes:

Desde la relación Servidor-Cliente el productor de contenidos (Service Provider) obtendrá su información desde 3 fuentes:

1º- Una base de datos tipo MySQL donde se realizaran peticiones a través de JDBC.

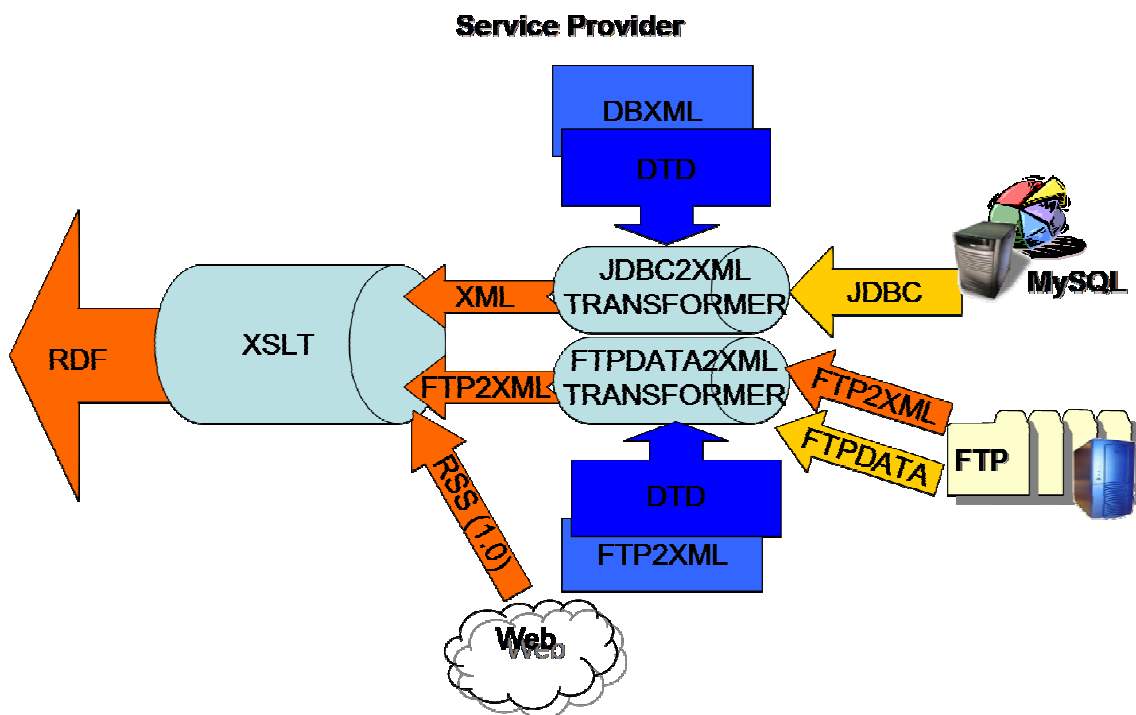
2º- Un FTP-Server de donde obtendrá los archivos disponibles para descargar por los clientes. La información sobre los archivos se obtendrá a partir de archivos XML (yo a este tipo de lenguaje lo llamo FTP2XML) que guardarán la información sobre el autor, una descripción sobre el archivo con posibilidad de incluir imágenes y el nombre del archivo que le aparecerá al usuario final. También se tendrá en cuenta la posibilidad que el administrador no quiera generar este tipo de archivos de descripción, por lo que la aplicación será capaz de generarlo por si sola.

3º- Tecnologías como RSS, para poder obtener las noticias de periódicos en tiempo real, de manera que el usuario podrá seleccionar que tipo de noticias y de que periódicos digitales obtenerlas.

Posteriormente todas estas fuentes serán utilizadas por el productor de contenidos (Service Provider), que se encargará de “traducir” toda esta información de distintos formatos a un formato único basado en XML y que transmitirá a nuestro Service Requester a través de una estructura tipo RDF. Para ello el Service Requester hará una petición usando el protocolo SOAP. Esta traducción se realizará en función de la fuente que, en caso de ser la base de datos, la información se traduciría a RDF; en caso de tratarse de una fuente de información procedente de Internet en forma de RSS (versión 2.0) la traduciría a RDF usando XSLT y por último cuando se tratara de una aplicación FTP pueden darse 2 situaciones:

1.- Que el administrador haya creado un archivo de descripción en xml (según indica la DTD de [FTP2XML](#)) del archivo almacenado en el FTP. En este caso usando hojas XSL lo convertiríamos en RDF.

2.- En caso de no haberse creado un archivo de descripción en xml, nuestra aplicación generaría dicho código xml para el archivo y continuaría el camino anteriormente descrito. El esquema se muestra en el gráfico siguiente:

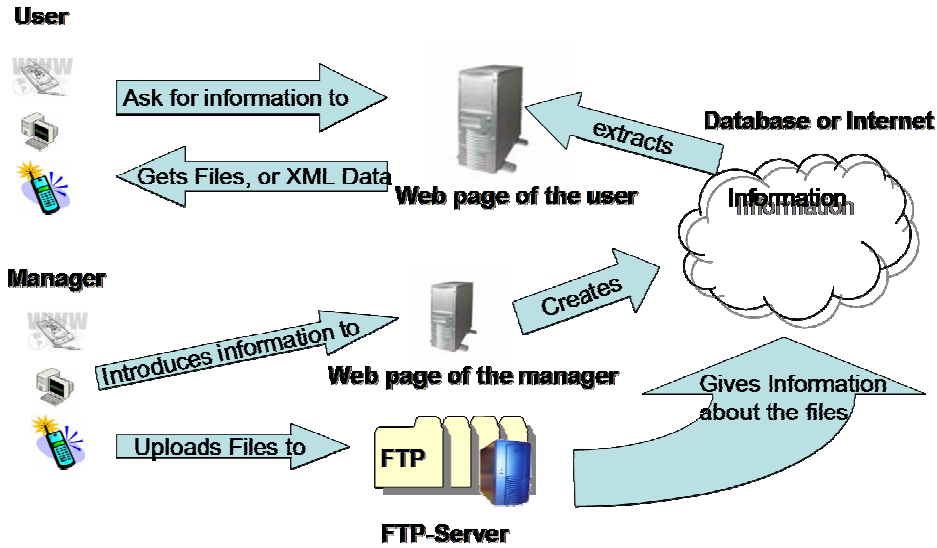


Una vez obtenida la información en RDF se encargaría el service requester de convertir dicha información en RDF a los diferentes estándares como SVG tiny (o basic), XHTML Basic usando CSS mobile para mostrar la página en dispositivos móviles o ya en HTML en el caso de tratarse de un navegador compatible.

Desde el punto de vista cliente-servidor, consistiría en hacer una petición http al service requester (a través de un servlet programado en java) donde, en el caso de que no dispusiese de ella, mediante el protocolo SOAP se le pediría al service provider

dicha información con lo que el service requester la obtendría desde la base de datos, FTP o desde Internet.

Desde el punto de vista del cliente, la estructura (obviando el funcionamiento interno de la aplicación) sería la que se muestra a continuación:



Por último cabe destacar que la complejidad de este diseño se debe a que persigue no tener que modificar el código interno de la aplicación para un simple cambio como pueda ser el del aspecto o la aparición de una nueva versión de RSS o similar, que únicamente nos obligaría a cambiar una hoja de estilos o una plantilla XSL en el prototipo que se propone.

Realizado por Antonio Quintáns Ocaña