Una familia de herramientas para la edición y publicación de noticias basada en NewsML

Ignacio García Rodríguez de Guzmán

Escuela Superior de Informática
Universidad de Castilla la Mancha
Paseo de la Universidad, 4
13071-Ciudad Real
Ignacio.garcia2@alu.uclm.es

Resumen. Presentamos en este paper una familia de aplicaciones que pretende facilitar la tarea de crear, diseñar, publicar y almacenar publicaciones periódicas. Por un lado, para la herramienta de creación del periódico, se propone utilizar la tecnología NewsML, basada en el estándar abierto XML, mediante la que representaremos cualquier tipo de información susceptible de ser publicada en un periódico electrónico. NewsML es un entorno estructural extensible y flexible para las noticias, que además soporta la representación de los elementos que pueden contener las noticias, las relaciones entre ellos y los metadatos asociados. El almacenamiento y organización de todos los documentos generados en NewsML se realizará mediante una base de datos orientada a XML. Este tipo de bases de datos nos ofrece toda la potencia necesaria para poder mantener de forma eficiente todas nuestras publicaciones, permitiéndonos un acceso totalmente transparente a los documentos en cualquier momento y facilitando la recuperación y reutilización de los contenidos. Por otro lado, la segunda herramienta consiste en una aplicación, que residiendo en el servidor, atenderá las peticiones realizadas mediante el protocolo de transporte http, accediendo a la base de datos en XML para montar una página web que mandará al usuario que realizó la petición.

Palabras clave: NewsML, XML, publicación periódica, periódico electrónico

1.- Introducción

Tradicionalmente, el periódico ha sido un excelente medio de información que ha permitido a todo el mundo la posibilidad de saber y conocer más allá de sus fronteras

geográficas. Hoy en día, las tecnologías de la información han puesto a disposición de las personas un análogo del antiguo periódico, pero esta vez en formato digital.

Es un hecho indiscutible el que Internet se ha imbricado en nuestras vidas pasando a ser en muchos casos un medio más para realizar muchas de nuestras actividades diarias, entre ellas, la de satisfacer nuestra necesidad de información. No es un hecho desconocido, aunque sí relativamente reciente, el de la existencia de periódicos publicados a través de la red. Estos periódicos (igual que sus análogos en papel) nos muestran la actualidad más reciente día a día, incluso a intervalos de tiempo más pequeños. Los periódicos electrónicos, como forma "evolucionada" del periódico tradicional, presentan una serie de mejoras indiscutibles como es la mejora de presentación de la información, disponiendo de fuentes mucho más ricas que los simples artículos y fotografías. Gracias a la naturaleza de este tipo de publicaciones, dispondremos de la información en modo de vídeo y audio entre otros.

Por la naturaleza de estos periódicos electrónicos, no podemos seguir los métodos tradicionales para realizar las publicaciones. La realización de una edición requerirá de personal específico con buenos conocimientos en las nuevas tecnologías. Es por esto, que surge la necesidad de tener una herramienta que de alguna forma pudiera automatizar la realización de las ediciones y mantener un repositorio o hemeroteca con todas las ediciones anteriores para posibles consultas.

Éste es el propósito de la familia de aplicaciones que vamos a presentar. Hay que decir en primer lugar, que este proyecto está en proceso de realización, pero siguiendo, como no, todos los puntos que expondremos. Mediante el uso de las aplicaciones que compondrán esta familia de herramientas, permitiremos a los responsables de la edición de los periódicos, que sin que tengan conocimientos fuera de los necesarios para la gestión de la edición de un periódico, puedan sacar a la red sus publicaciones de manera rápida y sencilla.

Se ha elegido el lenguaje *NewsML*, lenguaje de marcado de propósito específico basado en XML, para la representación de la información que va a contener una edición. NewsML dispone de una sintaxis específica para poder representar una publicación electrónica en su totalidad, desde el título de la misma, hasta las fuentes de los documentos incluidos. El hecho de utilizar un lenguaje basado en XML nos da portabilidad e independencia de plataforma, ya que XML está creado a partir de estándares [1]. NewsML es una tecnología relativamente joven, ya que su primera versión data de octubre del 2000, y hay pocos trabajos hechos, por lo que podemos decir que esta herramienta será, en parte, pionera en este tipo de trabajos.

2.- Trabajos relacionados

Desde hace algún tiempo, existen trabajos relacionados con el tema de la generación de periódicos electrónicos. Un ejemplo de ellos es [2], donde se presenta un sistema que ofrece también una buena gestión para el desarrollo de periódicos electrónicos. Sin embargo, discrepamos en ciertos aspectos con su sistema (aunque para las noticias usen también un lenguaje basado en XML, aspecto con el que sí estamos de acuerdo), por ejemplo, en su aplicación, el editor que se ofrece a los periodistas, facilita la manipulación de marcas de HTML (como se desprende de la explicación) para la edición de las noticias. Quizá no debería forzarse a los usuarios de la aplicación a

utilizar un lenguaje que, casi con seguridad, será desconocido para ellos. En nuestro sistema, también usamos un editor para la introducción de las noticias, pero el periodista sólo debe preocuparse de teclear el texto e introducir alguna propiedad de formato como puede ser la letra en negrita, cursiva o subrayado. El formato introducido por los periodistas sobre el texto se representará luego en la Web mediante hojas de estilo XSL, y de esta forma, no desviamos la tarea del periodista, que es la de centrarse en la redacción de las noticias.

Tanto en este sistema como en otros, se habla de código HTML como medio para representar la información, de HTML dinámico concretamente para facilitar la interacción al usuario con la Web, pero quizá no se tiene en cuenta que el HTML mezcla formato y contenido en un mismo documento. Como es bien sabido, el hecho de que se mezcle el formato y el contenido no es adecuado para representar un tipo de medio en el que el contenido y la información es el núcleo de nuestro servicio. Es por ésto que en nuestro sistema, utilizamos un lenguaje basado en XML, NewsML en particular, que se centra exclusivamente en el contenido, tratando la representación totalmente aparte. Otro inconveniente que se nos presenta al usar HTML dinámico, es que ese dinamismo es adicional al estándar HTML, por lo que no es necesariamente soportado por todos los navegadores y plataformas con acceso a la Web. En estos días, esto último es difícil que ocurra, pero es un hecho bien cierto que a veces pasa. No obstante, sí que existen multitud de trabajos relacionados con el tratamiento de documentos en XML, como los expuestos en las Segundas Jornadas de Bibliotecas Digitales, entre los que podemos destacar un sistema de suscripción para noticias digitales [3] y una propuesta de sistema de integración basado en XML [4].

Trabajos posteriores del mismo grupo de trabajo que realizó [2], nos muestran lo que pudo ser una versión más avanzada del sistema que presentaron en un inicio en [14]. Este artículo muestra un sistema con cuyos propósitos nos identificamos más, ya que el proyecto que tenemos intención de realizar sigue en cierto modo la línea de su caso de estudio, ya que coincidimos en algunos de los apartados que más adelante expondremos.

En cualquier caso, no hemos encontrado referencias a trabajos que utilicen NewsML para la implementación de publicaciones electrónicas. Sí diversos ejemplos sobre la utilización del mismo, pero no ningún sistema que lo realice de forma automática, que es lo que pretendemos.

3.- NewsML como modelo de representación.

Se ha elegido NewsML porque es un lenguaje de propósito específico, que está creado exclusivamente para crear publicaciones de carácter electrónico. NewsML representa un marco compacto, extensible y flexible para las noticias basado en XML y en otros estándares. Este lenguaje soporta la representación de noticias electrónicas, colecciones de noticias, las relaciones que pueda haber entre ellas y sus metadatos asociados. Por su naturaleza y origen del XML, soporta múltiples representaciones de la misma información [5], [6].

Podríamos resumir en tres las características de NewsML:

→ NewsML provee un marco para la gestión e intercambio de noticias.
→ NewsML está basado en XML.
→ NewsML es independiente de los medios que pueda contener, existentes o por inventar.

La estructura básica de un documento NewsML estaría compuesta por los elementos representados en la siguiente figura:

Fig. 1. Modelo en árbol. Near & Far[7]

donde el componente más importante podríamos decir que será el *NewsItem,* que es la sección donde se ubicará el contenido de la noticia. A continuación, mostraremos en la figura 2 las partes en las que se desglosa esta sección *NewsItem:*

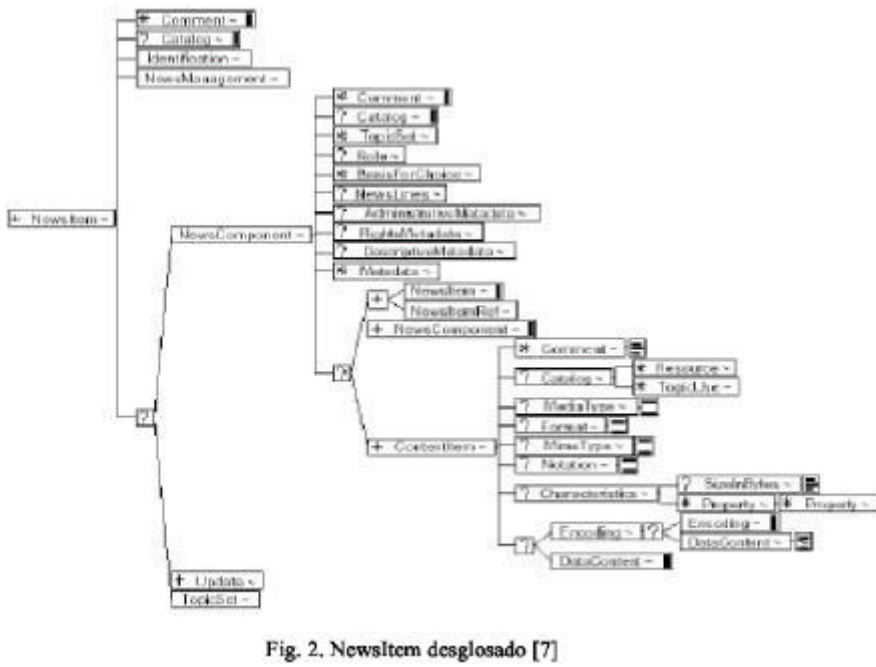

Fig. 2. NewsItem desglosado [7]

Es importante citar un toolkit, el *NewsML Toolkit [12],* que nos provee una interfaz de programación sencilla para crear documentos en NewsML. La versión utilizada de este toolkit, diseñado para Java, utiliza el DOM como medio de acceso, gestión y creación de los documentos. Sin extendernos en demasía, citaremos una

característica de este toolkit, que es la de permitirnos realizar test sobre los documentos que vamos construyendo para comprobar su corrección mediante numerosos casos de prueba ya implementados con el paquete de pruebas unitarias JUnit [13].

4.- Arquitectura básica del sistema.

En este apartado vamos a hacer un estudio más detallado de la composición, arquitectura y servicios de la familia de herramientas para la edición y publicación de noticias.

4.1.- Aplicaciones.

Esta familia de herramientas se compone básicamente de dos herramientas. La primera de ellas nos dará soporte para gestionar el proceso de diseño y creación del periódico electrónico; y la segunda, es la encargada de "montar" una página web con los contenidos del periódico cuando un usuario vía Internet accede a la dirección del mismo.

Cuando se pensó en esta herramienta, se pensó en que fuera capaz de dar la mayor funcionalidad dentro del alcance del proyecto, y que a la vez fuera lo más sencilla posible de utilizar. Esta aplicación nos va a permitir crear una edición personalizada del periódico, tanto en presentación como en contenidos.

Gracias a esta aplicación, vamos a poder decidir la estructura completa de nuestro periódico, comenzando por las secciones que vamos a incluir y los contenidos de las mismas.

4.1.1. Primera herramienta: Gestión de la edición de la publicación electrónica.

Como todos sabemos, y a veces sufrimos, la tecnología avanza a un ritmo vertiginoso, y lo que hoy es un descubrimiento, mañana estará obsoleto, simbólicamente hablando. Por esta razón, el diseño y desarrollo se ha planteado de forma que pueda ser fácilmente adaptable a los nuevos cambios tecnológicos. A continuación pasaremos a comentar ésto, ya que es un punto fuerte en el diseño arquitectónico de la aplicación.

El lenguaje de programación utilizado para desarrollar esta familia de aplicaciones ha sido Java de Sun MicroSystems [8]. Pero hemos de distinguir entre el lenguaje con el que se desarrolla la aplicación del lenguaje con el que representamos la información del periódico, NewsML en nuestro caso.

Dado que la tecnología esta evolucionando a tal velocidad, debemos tener cuidado a la hora de comprometernos con el lenguaje en el que se basará nuestro producto, el periódico electrónico. Por ello, a pesar de estar totalmente convencidos en el uso de NewsML como modelo de representación de la información, nos hemos curado en salud y hemos diseñado la aplicación de forma que una nueva versión o

revisión de NewsML que incluya modificaciones o nuevas funcionalidades pueda ser adaptada sin que suponga un cambio traumático en la aplicación.

Para el desarrollo de aplicación hemos utilizado una arquitectura multicapa en tres niveles:

→ Nivel de presentación o nivel superior, concierne todo lo relacionado con la interfaz.
→ Nivel de dominio o nivel intermedio, contiene toda la lógica de negocio de la aplicación, es el punto más importante.
→ Nivel de almacenamiento o nivel inferior, este punto será ampliamente desarrollado más adelante. Concierne al almacenamiento de los documentos generados.

Es en el nivel de dominio, donde hemos "tomado precauciones" para evitar las consecuencias derivadas de los cambios de tecnología de representación. En nuestro caso, la tradicional capa de dominio se divide en dos capas a su vez, como podemos ver en la figura 3:

→ Dominio dependiente de la aplicación.
→ Dominio dependiente de la tecnología de representación de la información.

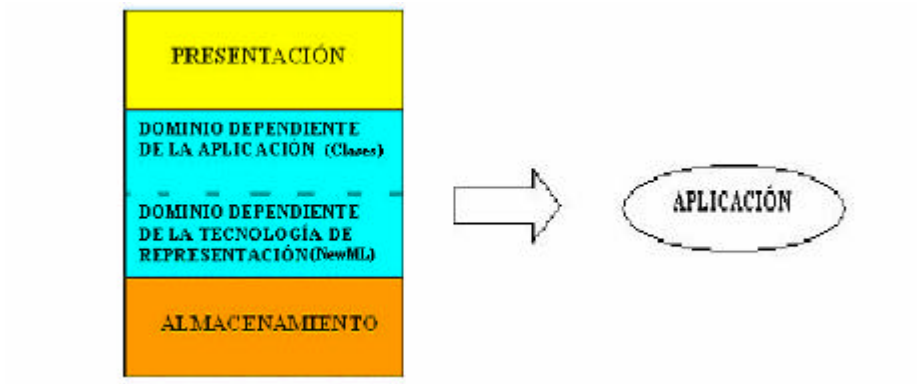

Fig. 3. Arquitectura multicapa de la aplicación.

La capa de *"dominio dependiente de la aplicación"* es la que va a modelar la lógica de la aplicación. Supuesto que con esta aplicación tenemos que realizar la gestión del diseño y desarrollo de un periódico electrónico, en esta capa estarán definidas todas las clases que modelan este comportamiento. Teniendo en cuenta que la estructura de un periódico electrónico va a ser bastante estática, estas clases no sufrirán modificación alguna a lo largo del ciclo de vida de este software. Sin embargo, no hay nada más variable que la tecnología, y ésta es la razón de ser de la capa de dominio dependiente de la tecnología. En la capa de dominio dependiente de la aplicación, crearemos la edición del periódico electrónico que vamos a publicar, pero la crearemos a nivel lógico, a nivel de objetos, en instancias de las clases que representan la estructura de un periódico.

La capa de dominio *"dependiente de la tecnología de representación"* contiene las clases que se encargan de generar los documentos en NewsML a partir de las clases antes mencionadas. De haber enfocado ésto de forma global, un cambio en la versión de NewsML, o incluso, en el modelo de representación, trastocaría todo este nivel, haciendo necesario un nuevo diseño e implementación. Gracias a esta separación, tan solo tendremos que modificar algunas de las clases de ese dominio dependiente de la tecnología de representación para seguir generando documentos válidos. Podríamos incluso extender nuestra aplicación de forma que generáramos periódicos electrónicos en distintos lenguajes según nos convenga, sin que para ello tuviéramos que modificar el resto de la aplicación. En la figura 4, podemos ver como la capa de dominio *dependiente de la aplicación* se comunica con la capa de dominio *dependiente de la tecnología*, pero no depende la implementación de una, de la implementación de la otra, por lo que, como hemos dicho, podríamos sustituir la segunda capa de dominio sin tener que tocar la capa dependiente de la aplicación.

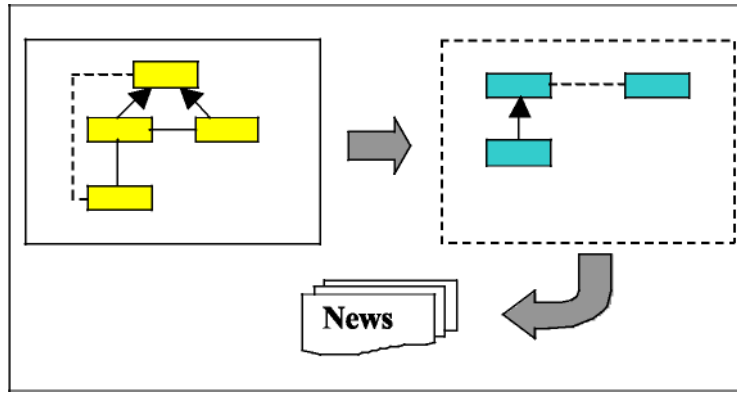

Fig. 4 Relación existente entre las dos capas del nivel de dominio.

4.1.2. Segunda herramienta: Distribución de la publicación electrónica.

Una vez que hemos diseñado y desarrollado nuestro periódico electrónico personalizado, llega el momento de publicarlo, y esa es la tarea de la segunda aplicación de la familia de herramientas sobre la que trata este paper.

Está herramienta, más sencilla que la primera, es básicamente una aplicación que correrá en el servidor en el que tengamos los documentos. Cuando un usuario desee acceder a los contenidos de nuestro periódico, simplemente accederá desde su navegador escribiendo la dirección de la web. Cuando la aplicación servidora reciba la petición, accederá a la base de datos, recuperará los documentos relacionados y montará una página web que devolverá al usuario.

La aplicación no sólo resolverá todas las peticiones que los usuarios realicen, sino que además permitirá la consulta de ediciones anteriores o documentos específicos, siempre y cuando todavía existan y su acceso este permitido. De esto último podemos deducir que el sistema que nos ocupa no solo será una herramienta de

diseño, edición y publicación de periódicos electrónicos, sino que también hará las veces de repositorio de documentos ya publicados. Las posibilidades y la potencia que nos ofrezca este repositorio vendrán dados no solo por las características intrínsecas de NewsML, si no por todas las posibilidades que nos ofrece el uso de una base de datos orientada a XML.

En la figura 5, podemos observar el papel que juega la aplicación servidora dentro de nuestra familia de herramienta, como hace de nexo de unión entre el entramado de documentos NewsML y las hojas de estilo residentes en el servidor, y el impaciente usuario que desea leer el periódico desde su hogar.

Como se puede observar en la figura, en la base de datos de XML existen dos tipos de documento: por un lado tenemos los documentos en NewsML, y por otro los documentos XSL [1] u hojas de estilo. Es aquí donde queremos hacer hincapié en esa independencia de contenidos y representación que sería tan importante para un sistema de este tipo. Las hojas de estilo, van a ser el medio que vamos a tener para formatear nuestro documento final, el que se presentará al usuario. En los documentos en NewsML se expondrán los contenidos del periódico, y esos contenidos quedarán en la base de datos mientras sea necesario, pero la presentación puede variar con el tiempo, bien por que los documentos queden obsoletos, bien por que decidamos cambiar la apariencia general de nuestro periódico, sin embargo, en cualquiera de los dos casos, tan solo habrá que modificar o reponer los documentos XSL. Si utilizáramos un lenguaje de marcado que no separara el contenido de la representación, hacer esta distinción nos sería imposible, y cualquier tipo de cambio en la presentación podría suponer una pérdida de los contenidos, por no hablar de que en ese caso, el contexto del documento no nos ofrecería ninguna ayuda a la hora de realizar búsquedas de algún tipo de contenido.

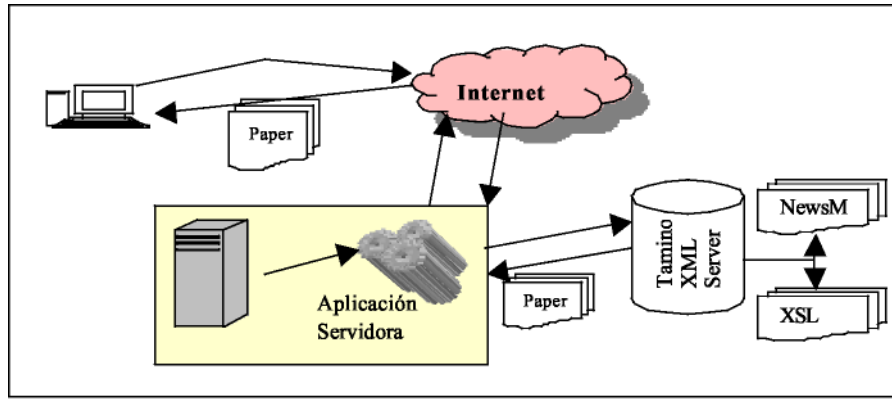

Fig. 5 Funcionamiento de la aplicación servidora

4.2. Almacenamiento de los documentos existentes.

En el ámbito de esta aplicación, al asunto del almacenamiento es de vital importancia, ya que del tipo de gestión que hagamos, dependerá en gran medida la disponibilidad y mantenimiento de los documentos.

El hecho de tratar directamente con documentos en XML, nos obliga a realizar un tratamiento especial de los mismos. Resulta muy complejo almacenar documentos XML (y, por tanto, NewsML) en bases de datos relacionales, por lo que otros autores han buscado mecanismos de almacenamiento y recuperación más eficientes, como bases de datos objeto-relacionales [9]. Hay que comenzar considerando el hecho de que un documento NewsML tiene una estructura jerárquica que debe ser respetada. Esta estructura jerárquica nos es imprescindible para tener un orden dentro del documento y un control sobre los contenidos entre otras cosas. Si consideramos el almacenar un documento de las características estructurales de NewsML, a primera instancia se nos presentan dos alternativas, bien como una cadena de texto enorme, o bien intentando emular la estructura del documento mediante tablas e integridades referenciales entre ellas. Si optamos por la primera opción, ya podemos olvidarnos de estructura jerárquica, porque lo único que tendremos será una secuencia de caracteres, sin mencionar cualquier tipo de enlace que haya de un documento a otro, que quedaría totalmente anulado, por lo que cualquier tipo de búsqueda de un documento podría llegar a ser casi imposible. Si nos decidimos por implementar una serie de tablas y relaciones entre ellas para emular la estructura jerárquica del documento, habría entonces que ponerse a pensar la cantidad de tablas que podrían hacer falta para representar un documento en NewsML no muy grande, sin ni siquiera tener en cuenta cuando podemos encontrar recursión en alguna etiqueta del documento. Otra gran desventaja que ofrece esta segunda alternativa es la de que para cada tipo de documento que queramos almacenar, tendríamos que rehacer el esquema de la base de datos, o modificar el existente, lo cual puede ser una tarea muy dura de llevar a cabo.

Por este motivo se impone la necesidad de tener que usar un medio que nos permita gestionar de una manera eficiente los documentos, y a la vez nos permita usar todas las ventanas propias del lenguaje XML. Además, para poder guardar documentos basados en XML, sólo necesitaremos disponer del esquema del lenguaje basado en XML, o bien del DTD, que puede ser pasado automáticamente a esquema.

La opción seleccionada para llevar a cabo el almacenamiento de los documentos, es la de utilizar una base de datos basada en XML, Tamino XML Server de Software AG [11]. En esta base de datos, no vamos a utilizar tablas propiamente dichas, sino que mantendremos los documentos en colecciones cuyas propiedades quedan definidas por los DTDs o los esquemas de los documentos que serán almacenados.

Al igual que en cualquier base de datos, en Tamino también se permite hacer consultas sobre su contenido (característica vital de la que hará uso intensivo la aplicación servidora de publicación), y para ello, aprovecha las capacidades exclusivas de XML, como lo es XPath [10] . Para procesar los datos, se utilizarán mecanismos estándares de XML como lo son [5] SAX, DOM, XSLT, XLink, ...

Como los modernos gestores de bases de datos tradicionales, Tamino ofrece también la posibilidad de integrar otros tipos de contenidos aparte de los documentos en NewsML, como lo pueden ser los archivos de audio, vídeos, y otro tipo de medios susceptibles de ser incluidos en una publicación electrónica.

Esta base de datos va a constituir también el nexo de unión de las dos aplicaciones que forman parte de nuestra familia de herramientas. Podríamos considerar a la herramienta de edición como una aplicación productora, y a la

aplicación servidora como una aplicación consumidora de documentos NewsML y de las hojas de estilo que darán formato al periódico electrónico (figura 4):

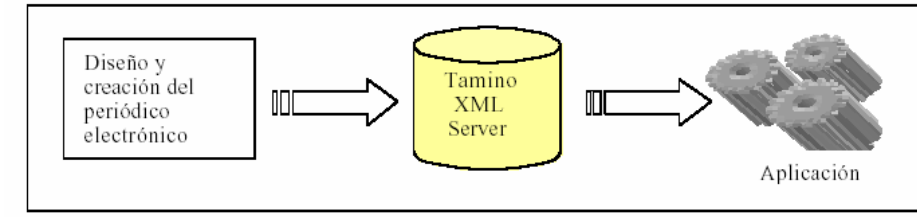

Fig. 4 Relación entre los componentes del sistema.

5. Conclusiones y Trabajos Futuros

El resultado de este trabajo será una completa herramienta que ofrecerá una cobertura total en la gestión de las publicaciones electrónicas. Dada la arquitectura que se ha planteado para la herramienta editora, podremos adaptar nuestro sistema a nuevos estándares y tecnologías de una forma sencilla y no muy costosa.

Un objetivo futuro de esta herramienta, ya que por ahora el alcance del proyecto esta limitado, es el de implementar un cliente de edición de noticias sobre la Web para permitir a los periodistas que no estén en la redacción, que puedan incluir sus noticias en las ediciones diarias. Estas podrían ser recogidas y procesadas mediante Servicios Web y protocolo SOAP.

6. Referencias

[1] Gutiérrez, A., Martínez, R. XML a través de ejemplos. (2001). Editorial Ra-Ma.
[2] Luque, V. Disponible en 16/02/2001: *Concepción y desarrollo de un periódico electrónico personalizado*. http://www.rediris.es/rediris/boletin/46-47/ponencia5.html
[3] Pérez, J.M., Alfaro, I, Aramburu, M.J., Berlanga, R. Sistema de Suscripción basado en XML para noticias digitales. Segundas Jornadas de Bibliotecas Digitales, Noviembre 2001.
[4] Manzano, J.C., Polo, A., Salas, M., Arévalo, L. SIX: Una propuesta de sistema de Integración basado en XML. Segundas Jornadas de Bibliotecas Digitales, Noviembre 2001.
[5] http://www.w3.org/Style/XSL/
[6] http://www.iptc.org/site/NewsML/specification/newsmlfunctionalspecv1.05.html
[7] http://www.iptc.org/site/NewsML/specification/NewsMLv1.0.treeview.pdf.
[8] http://.Java.Sun.com
[9] Aramburu MJ, Berlanga R, Llidó D y García S. Almacenamiento y recuperación de periódicos digitales.
[10] XML Path Language (XPath): http://www.w3.org/TR/xpath
[11] http://www.softwareag.com/tamino
[12] http://newsml-toolkit.sourceforge.net/
[13] http://www.junit.org/
[14] Diseño de un Periódico Electrónico con XML. http://www.it.uc3m.es/~per/doc/cnit/cnit.html