

BIBLIOGRAFÍA

- HTML (2002). Manual de uso de HTML. <http://www.lawebdelprogramador.com/>
- ISAGRI (1998). Isalact: Programa de gestión de explotaciones de vacuno de leche.
- ISAGRI (2003). Isacab: Programa de gestión de explotaciones caprinas ("DEMO").
- GONZÁLEZ REDONDO, P. (2001). ISALACT. Manual de Usuario. Departamento de Ciencias Agroforestales. EUITA. Universidad de Sevilla. 53 págs.
- NANTA (1997). Kompas Ovejas: Programa de gestión de explotaciones de ovino. Madrid.
- SOKKIT. (2002). MySQL Reference Technical Reference. <http://www.sokkit.net>
- RAMOS, A. (1996). Informatización de registros de los parámetros técnicos de una explotación caprina en el valle del Guadalquivir
- STIG SHAETHER BAKKEN (2003). Manual de uso de PHP. <http://www.php.net>

UN SISTEMA DE AULA VIRTUAL QUE COMBINA SISTEMAS DE HIPERMEDIA CON APLICACIONES DE VIDEOCONFERENCIA

Jóse Antonio Ortega Ramirez, José Manuel Márquez Márquez,
Jesús Torres Valderrama, M^a José Escalona Cuaresma
Dpto. de Lenguajes y Sistemas Informáticos
Facultad de Informática y Estadística
Universidad de Sevilla

RESUMEN

La idea original de este trabajo de investigación era el desarrollo de un sistema de aula virtual que combinase de manera precisa los sistemas hipermedias docentes con las aplicaciones de videoconferencias, combinando apropiadamente los aspectos docentes e informáticos en su implementación.

Diseñar e implementar sistemas multimedia para la educación no es una tarea sencilla. No sólo por el trabajo que conlleva sino por el cuidado y la dedicación que hay que prestar por encima de todo al diseño de la aplicación, de forma que la función pedagógica de la misma no se vea mermada lo más mínimo por un diseño poco funcional que dificulte el aprendizaje del alumno.

En este artículo mostraremos un conjunto de estrategias (patrones de diseño) apropiadas para diseñar y construir aplicaciones de hipermedia, y algunos ejemplos de diseño efectivo con patrones en un sistema multimedia para la educación basado en ventanas.

ABSTRACT

The idea of this investigation work was the development of a virtual classroom system that combines an educational hipermedia systems with videoconference applications. This merging may be carry out combining the educational and computer aspects appropriately.

To design and to implement educational multimedia systems is not a simple task. It is necessary besides working that it takes care the design of the application, so that its pedagogic function is not shrunk by a improper functional design that hinders the student's learning.

In this paper, we propose a set of strategies (design patterns). They are appropriate to design and to implement hipermedia applications. It is also included, some examples of effective design with patterns in an educational multimedia system based on windows.

INTRODUCCIÓN

Habitualmente el desarrollo de Sistemas Hipermedia suele hacerse utilizando directamente herramientas de autor a nivel de implementación, descuidándose el proceso previo de análisis y diseño de los aspectos estructurales, de navegación y de interfaz con el usuario que, de esta forma, no quedan recogidos en documentación alguna, haciendo complicado el mantenimiento futuro del sistema. Sin embargo, en los últimos años existe una tendencia a considerar el desarrollo hipermedia con un enfoque de proceso de ingeniería (del software), por lo que ya se han propuesto diferentes metodologías como RMM (Isakowitz y otros 1995), HDM (Garzoto y otros, 1993) o OOHDM (Schwabe y otros, 1996), que establecen la necesidad de considerar un diseño previo a la construcción del sistema y ofrecen una serie de técnicas, más o menos formales, para recoger en diferentes modelos abstractos las especificaciones del sistema hipermedia a desarrollar.

Han aparecido metodologías que asumen la orientación a objetos como paradigma de diseño al igual que ha ocurrido en el ámbito de la Ingeniería del Software. En el caso hipermedia, el enfoque Orientado a Objetos en los modelos es muy útil debido al gran nivel de abstracción que ofrece y a sus mecanismos de composición (generalización, clasificación y agregación) que facilitan el modelado de la estructura hipermedial (Rossi y Schwabe) (Garrido y otros 1997). Esto queda reflejado en las dos metodologías mas conocidas de desarrollo orientado a objetos de sistemas hipermedia como son EORM y OOHDM, así como en los trabajos (Escalona 2002). En ambos casos el proceso de desarrollo comienza con una fase de análisis de la información del sistema en la que se aplica alguna de las metodologías de orientación a objetos conocidas en ingeniería del software, como OMT (Object Modeling Technique).

Por tanto, el diseño y desarrollo de contenidos hipermedia está ya bien difundido y estudiado en nuestros días, llegando determinados autores (Lyardet y otros, 1998) a distinguir tres grupos importantes correspondientes con tres decisiones que debemos adoptar en cualquier diseño de un sistema hipermedia para educación:

- Cómo organizamos la información: Según organizamos la información en temas, en unidades, por palabras claves...
- Cómo organizamos la navegación a través de la información: ¿Cómo se desplazará el usuario para acceder a distintas informaciones? ¿Guarda relación la forma de navegar con el tema en el que se encuentre en un momento determinado? ¿Qué tiempo de adaptación necesitará el usuario para esa nueva interfaz de navegación?
- Cómo implementamos el sistema: Según la información esté mapeada en páginas, o contenida en una base de datos, posee sistemas de minería de texto o de datos (text mining o data mining) que ayuden a extraer ciertas informaciones, si será un sistema interactivo o no, etc.

Con cada una de estas decisiones podemos asociar una serie de patrones que nos facilitan la tarea del diseño, con el objetivo de construir sistemas multimedia educativos de calidad. Los patrones de diseño aparecieron para la arquitectura (Alexander y otros, 1977) y han sido adaptados con éxito en el diseño de software (Gamma 1995). Describen problemas que suceden repetidamente, así como la solución a esos problemas, de forma que podremos usar esa solución tantas veces como necesitemos en diferentes aplicaciones y contextos.

El diseño inicial debe conjugar una serie de principios psicopedagógicos con el fin de conseguir la mayor optimización didáctica posible del sistema. Hay que tener en cuenta cualidades tan importantes como la amigabilidad, comprensibilidad, navegabilidad, actualidad o interactividad (Astigarraga y Carrera, 1999), evitando siempre caer en el barroquismo, que no suele contribuir a otra cosa que no sea la distracción, llevando en algunos casos a ser la principal causa del llamado analfabetismo funcional.

En este artículo vamos a presentar un conjunto de patrones que son especialmente aplicables al diseño de aplicaciones para la educación. Nótese que estos patrones pueden aplicarse conjuntamente de forma conjunta, no siendo, por tanto, excluyentes entre sí. Como ejemplo vamos a mostrar el uso de estos patrones en Eduka (Marquez 2002) presentado de manera más extensa en (Ortega 2002). Se trata de un sistema para educación semipresencial realizado íntegramente en Java. Comentaremos algunos patrones de diseño ya introducidos en (Lyardet y otros, 1998) y su aplicación a este sistema, diferente a lo que estamos acostumbrados a ver en Internet, puesto que a diferencia de la mayoría de aplicaciones en las que los nodos de información se presentan en forma de páginas html, en Eduka lo hacen en ventanas, puesto que se ha diseñado como un sistema basado en navegación por ventanas.

NAVEGACIÓN A TRAVÉS DE LA INFORMACIÓN

Los patrones de esta sección, encaran problemas relativos a la organización y acceso a la información. Los patrones agrupados en esta categoría nos ayudarán a identificar cuestiones tales como la unidad mínima de presentación de información, separar mecanismos de navegación del propio contenido educativo, evitar la sobrecarga de cognitiva del usuario, comprensibilidad del conjunto, etc. que tan importantes son en la fase de diseño de un sistema multimedia para la educación.

Vamos a presentar patrones descubiertos y usados en el contexto del desarrollo de aplicaciones multimedia para la educación.

Nodo como unidad simple de información

Objetivo: ¿Qué constituye una unidad de información con respecto a la navegación?. ¿Como dividimos toda la información disponible de forma que su presentación al usuario sea estructurada, cómoda y eficaz? Para solucionar estos puntos, aplicaremos la filosofía de este patrón.

Desarrollo: Un nodo de información deberá constituir por sí mismo un conjunto de información con sentido propio para un conjunto de usuarios que llevan a cabo una tarea concreta. Con frecuencia, cuando la cantidad de información disponible para un tópico dado es alta, habrá que sopesar el beneficio que supone presentar tanta cantidad de información de una sola vez, la efectividad de obtener dicha información en relación con la navegación involucrada en el proceso, así como le sobrecarga cognitiva que supone llegar a ella y qué inconvenientes plantearía subdividir el nodo en otros nodos independientes más pequeños, si eso es posible.

En muchas ocasiones, debido a la reducida cantidad de información que podrá ser presentada al mismo tiempo, la unidad simple de información a la que llamamos hoja, quedará subordinada a ese límite.

Es importante hacer notar que la organización de la información no debería verse perjudicada por como será mostrada al usuario.

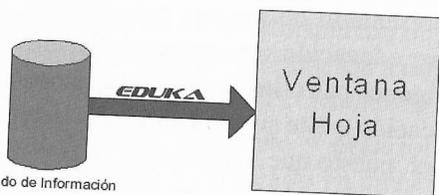


Figura 1. Unidad mínima: Nodo

Nuestra aproximación: Reunir toda la información relativa al mismo tópico en un mismo nodo de información. Separar de forma clara la navegación de los diferentes tópicos que dan lugar a nodos de información, abstrayendo de esta forma la interfaz de la información.

Uno de los aspectos más difíciles de este patrón es definir claramente dicha separación. Diseñar nodos como unidades conceptuales es importante porque simplifica el mantenimiento

Contexto Navegacional

Objetivo: ¿Cómo organizar la información? ¿Se deben establecer relaciones entre el estado de navegación y la forma en que puede ser presentada la información? ¿Es bueno proporcionar varios métodos de acceder a la información o es mejor "guiar" al usuario por un camino fijo y determinado por la aplicación? En un sistema destinado a la educación no es aconsejable dejar a la elección del usuario el camino a seguir para acceder a determinados nodos de información, puesto que en la mayoría de las ocasiones es imprescindible seguir una ruta prefijada por el docente: para explicar el tema 2 es necesario haber explicado anteriormente el tema 1, puesto que en el tema 2 se hace uso de conocimientos desarrollados en el tema anterior. Sin embargo en un mismo tema pueden aparecer interrelacionados una serie de conceptos y si el orden en el que se acceda a cada uno de ellos no es relevante... ¿imponemos la restricción de orden? ¿Dejamos que el usuario pueda acceder desde uno de estos conceptos a cualquiera de los otros relacionados con él? Este patrón nos ayudará a solucionar este problema.

Desarrollo: A menudo surgen multitud de caminos que llevan a la misma información (mismo nodo de información). Por ejemplo, supongamos que disponemos de un sistema multimedia que nos relaciona los profesores de un departamento y sus áreas de investigación. Si buscamos por un determinado profesor, deberíamos llegar a un nodo de información sobre dicho profesor, que a su vez debería mantener relación con cada uno de los nodos de cada área de investigación en el que está interesado. Igualmente, si buscamos un área de investigación en concreto, debemos poder llegar al nodo de información de cada uno de los profesores que colaboran en ese área. Igualmente si nos interesamos en la búsqueda de información acerca de un departamento, se deberá mostrar toda la información relativa al mismo, y en ella deberán aparecer las áreas de investigación en las que participa el departamento y cada uno de los profesores. Vemos pues, que existen multitud de vías para llegar a una misma

información. Pero lo que no debemos es duplicar dicha información, sino plantear los mecanismos de navegación apropiados y las relaciones entre los distintos tópicos de información. En definitiva, crear relaciones entre nodos de información.

Nuestra aproximación: Definir características peculiares de los objetos, que enriquezcan la interfaz del objeto que está siendo visitado. El contexto navegacional esta compuesto de un conjunto de nodos de información y enlaces.

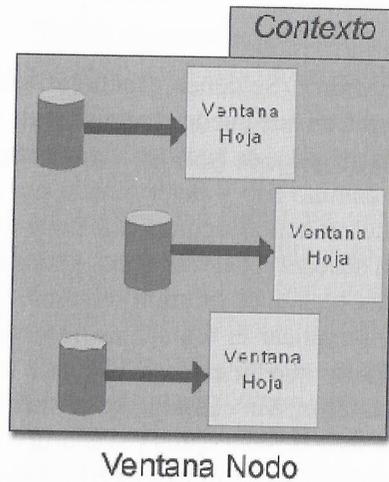


Figura 2. Ventana Nodo

Factoría

Objetivo: ¿Como podemos presentar la información que necesita el usuario de forma que ésta pueda ser correctamente comprendida? ¿Cómo poder presentar informaciones de nodos de información relacionados de forma eficiente?

Desarrollo: Este es un problema muy importante y común en sistemas multimedia para la educación. En numerosas ocasiones, es necesario hacer referencia a conceptos relacionados. El autor tiene entonces la posibilidad de añadir enlaces a los nodos relacionados, pero entonces, si existen muchos nodos estaríamos obligando al usuario a ir y volver continuamente de un nodo a otro para proseguir con la lectura, y esto redundaría en una mayor distracción del usuario y por tanto una menor efectividad del sistema.

Nuestra aproximación: La técnica usada para solucionar este problema pasa por mostrar de alguna forma la información relacionada en el mismo nodo. Esto se consigue fácilmente activando los nodos de información adicional dentro

Los nodos son decorados con información adicional sobre el contexto particular y enlaces adicionales. Es conveniente, en la medida de lo posible, separar los enlaces de la información, de modo que la información que se presente se parezca lo menos posible a una detestable colección de enlaces, que no sirve para otra cosa que para despistar al usuario y que hace tediosa e ineficaz la labor educativa del sistema. Por ello, y sobre todo en sistemas hipertextuales, es aconsejable combinar este patrón con el patrón Factoría que explicamos a continuación, para evitar el continuo ir y venir del usuario entre los enlaces.

del nodo actual, por ejemplo con ventanas pop-up (emergentes) que pueden ser desactivadas fácilmente, reduciendo así la sobrecarga de navegación, o bien reservando un espacio del nodo para estos menesteres.

Referencia Activa

Objetivo: ¿Como podemos proporcionar una referencia permanente sobre el estado actual de la navegación, combinando una herramienta de orientación con una forma fácil de navegar en un conjunto relacionado de nodos, en el mismo nivel de abstracción?

Desarrollo: En muchas aplicaciones se necesita proporcionar al usuario una forma de saber dónde se encuentra (qué es lo que tiene ante sí) y habilitarle para decidir qué nodo quiere visitar próximamente. La solución más corriente es incluir un índice de enlaces, pero esto requiere que el usuario vaya desde el nodo actual al índice para saber dónde se encuentra y desde allí ir a otro nodo, y así sucesivamente, aumentando la carga cognitiva del proceso y finalmente cansando y desorientando al usuario.

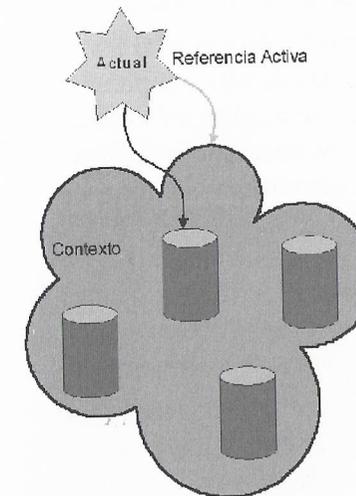


Figura 3. Referencia Activa

Nuestra aproximación: Una buena solución es diferenciar al índice y los nodos de información en objetos diferentes, ambos visibles al mismo tiempo. Una modificación del objeto que actúa de índice provocará la actualización del objeto que contiene al nodo de información, cargando el nuevo nodo.

ORGANIZACIÓN DE LA CAPA DE PRESENTACIÓN

Separación entre información e interacción

Objetivo: ¿Como diferenciar contenidos y varios tipos de controles diferentes en la presentación visual de nuestro sistema? La diferenciación entre controles que efectúan tareas diferentes debe ser clara, especialmente entre controles que reciben información por parte del usuario: controles de entrada,

mos llegar a ninguna otra ventana de información. Cada ventana hoja responde a un tópico concreto, por ejemplo, “propiedades de configuración de la transmisión de vídeo y audio en tiempo real” como se muestra en la figura 5.

Contexto navegacional

En Eduka se modela este patrón como “ventanas nodo” (no confundir con nodos de información). Las ventanas nodo agrupan un conjunto de ventanas hoja accesibles a través de una barra de menú. Estas ventanas hoja se abren en el interior de la ventana nodo facilitando así al usuario la asociación de los conceptos con el contexto en el que están relacionados, y posibilitando de este modo el tener más de una ventana nodo abierta con sus hojas en cada una, sin que por ello se produzca confusión alguna para el usuario. Al minimizar o cerrar una ventana nodo lo hacen igualmente todas las ventanas hojas contenidas en ella.

En la figura 6 se puede apreciar internamente dos ventanas hoja: propiedades predefinidas y ventana de monitorización

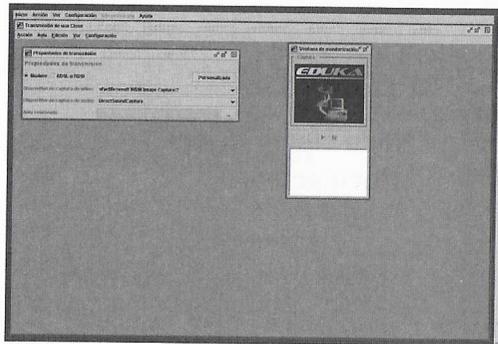


Figura 6. Ventana nodo de transmisión de una clase

cios de gramática, vocabulario, chat, lecturas,...) representan los contextos navegacionales, ordenados independientemente por temas o niveles que se asociarían con cada uno de los nodos de información de cada contexto.

Factoría

En Eduka no se encuentra implementado este patrón. Pero sí es importante tener en cuenta para los docentes que usen un sistema como Eduka a la hora de elaborar sus documentos o “apuntes” y distribuirlos. Algo parecido a la

filosofía de este patrón son los *tooltiptext* que nos facilitan información de un botón, imagen, etc con sólo pasar el ratón sobre él. Se debe hacer uso de esta utilidad para incorporar información relativa a las imágenes. Esto además aumenta la accesibilidad del sistema.

Referencia Activa

Eduka, en la ventana de consulta de la videoteca, permite que un usuario consulte las existencias de la videoteca. Es decir, de los ficheros multimedia que en ella se almacenan. Ahora bien, para facilitar la búsqueda de alguno en concreto, se añade un componente lista, que permite discriminar aquellos documentos que no pertenezcan a la selección. En concreto este componente permite seleccionar por nombre de asignaturas, visualizando únicamente los documentos referentes a la materia que busca el usuario. Además, si así lo desea puede seleccionar el campo “Todas” y se mostrarán todos los documentos existentes en la videoteca, sea cual sea la materia con la que esté relacionado. Este control lista nunca desaparece y siempre está activo mostrando el campo seleccionado, lo que permite al usuario saber en qué estado se encuentra actualmente y realizar nuevos accesos. Los nodos de información se crean dinámicamente en función del campo seleccionado en el control lista.

Como podemos apreciar, aún es posible refinar la ventana de consulta de la videoteca dividiendo el objeto que constituye la lista de documentos multimedia de la asignatura seleccionada, la página html, en dos objetos: un nuevo control lista que indexe cada documento y la propia página html con los enlaces a cada uno de ellos (que en este caso ya sería redundante). Esto nos evitaría la vuelta atrás para volver a seleccionar la misma asignatura en caso de que queramos acceder a dos documentos de la misma asignatura.

Selección de asignatura: Enlaces

Mostrar sólo los de la siguiente: Todas

Lista de archivos separados en la videoteca:

CODIGO	ASIGNATURA	TEMA	FECHA	ENLACE
1	ADP1	RISC vs CISC	2001-12-08 00:00:00	http://www.santa.es/seguro/1.html
2	ADP2	Computadores Inteligentes	2002-05-09 00:00:00	http://www.santa.es/seguro/2.html
3	DA1	Introducción	2001-10-01 00:00:00	http://www.santa.es/seguro/3.html
4	DA2	Tema 4	2002-07-26 00:00:00	http://www.santa.es/seguro/4.html

Eduka es un proyecto de la empresa Eduka, S.L. por José Manuel Martínez. Desarrollado en lenguaje Java y PHP. Escrito en lenguaje HTML y CSS.

Figura 7. Ventana de consulta de la videoteca

Separación entre información e interacción

En Eduka la interfaz de navegación la conforman las propias ventanas, decimos por ello que se trata de un sistema basado en ventanas. Los controles de salida se agrupan en la barra de menú de cada ventana y los controles de entrada, como combos, listas, campos de texto, etc forman parte del contenido de la ventana, respetando así la filosofía de este patrón. Los controles de la barra de menú nos llevan, como ya hemos comentado antes, a distintos nodos de información que pueden englobarse en el contexto que define la ventana. De esta forma hemos conseguido la separación que promulga este patrón.

Agrupación funcional

En Eduka hemos adoptado la medida de la separación de controles (botones principalmente), aumentando la separación entre controles de distintos grupos funcionales a 12 pixel y manteniendo en 5 la separación entre botones del mismo grupo funcional. Igualmente hemos creado menús de acciones, agrupando en cada menú las acciones con funcionalidad similar o que actúan sobre un aspecto común (texto, sonido, vídeo, mail,...). Igualmente hemos separado las funcionalidades de los distintos roles de usuario que se pueden tener en el sistema: administrador, profesor o alumno, creando menús que únicamente son accesibles por el rol que posea privilegios para ello.

En la figura 8, se puede observar en la parte superior como se aprecian 4 botones en dos grupos. Nótese la mayor separación entre el segundo y el tercero que entre los demás.

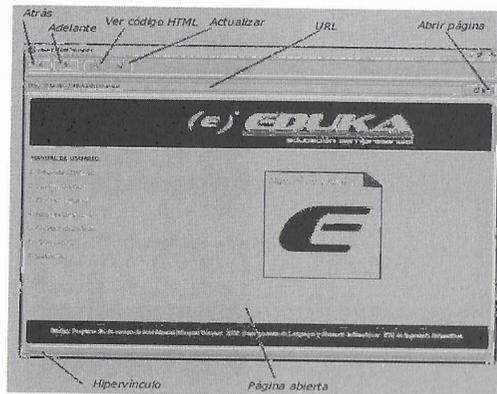


Figura 8. Navegador Web en Eduka

Información bajo demanda

Un ejemplo de aplicación en Eduka de este patrón puede verse en la visualización de la videoconferencia. En tiempo real y mientras se está recibiendo las tramas de audio y vídeo, el alumno puede: aumentar la ventana de recepción de vídeo, re-escalándose así la imagen (*zoom-in*), disminuirla (*zoom-out*), elegir recibir únicamente la señal de audio para ahorrar ancho de banda, elegir recibir únicamente la señal de vídeo o recibir las dos. Todo sobre el mismo nodo de información que constituye la videoconferencia en tiempo real.

CONCLUSIONES

Hemos conseguido facilitar la compresibilidad y el acceso de la información mediante un sistema multimedia complejo cuyo coste de desarrollo es muy alto en comparación con sistemas hipertextuales más tradicionales. Con la aplicación en el diseño de nuestro sistema del conjunto de patrones que se muestran en este artículo se ha conseguido un elevado nivel de eficiencia, una rápida familiarización de los alumnos con el sistema de navegación empleado, y una clara diferenciación entre navegación e información gracias a al concepto de ventana en sí.

Esperamos que con este trabajo los autores de aplicaciones multimedia para la enseñanza puedan ver un enfoque distinto de este tipo de aplicaciones y como los patrones de diseño puede ayudar no sólo a solventar sino también a identificar problemas que comúnmente se presentan en estas aplicaciones. Creemos que este tipo de sistemas puede compensar el coste de desarrollo con unos niveles muy altos de eficacia en la tarea educativa.

BIBLIOGRAFÍA

- ALEXANDER K., ISHIKAWA S., SILVERSTEIN M., JACOBSON M., FIKSDAHL-KING I. Y ANGEL S., (1997) *A Pattern Language*, Oxford University Press, New York.
- ASTIGARRAGA E., CARRERA X., (2000) Formación en tecnología básica mediante nuevas tecnologías. *Las Nuevas Tecnologías para la mejora educativa* (Cabrerero et al, 2000). Congreso Edutec 99. Editorial Kronos.
- ESCALONA M.J., MEJÍAS M., TORRES J. Y REINA A., (2002) Una técnica para el desarrollo de la navegación. *5º Workshop iberoamericano de Ingeniería de Requisitos y Ambientes Software*. La Habana (Cuba).

- ESCALONA M.J., MEJÍAS M. Y TORRES J., (2002) Getting requirements in Global Information Systems. *Object Oriented Information Systems*, Montpellier (Francia).
- GAMMA P., HELM R., JOHNSON R. Y VLISSIDES J., (1995) *Design Patterns: elements of reusable object-oriented software*. Ed. Addison Wesley.
- GARRIDO A., ROSSI G. Y SCHWABE D., (1997) Patterns Systems for Hypermedia *PloP'97, Pattern Language of Programming*.
- GARZOTTO F., SCHWABE D. Y PAOLINI P., (1993) HDM – A Model Based Approach to Hypermedia Application Design. *ACM Transactions on information Systems*, 11 (1), pp. 1-26.
- ISAKOWITZ T., STHOR E. Y BALASUBRAMANIAN P., (1995) RMM: A methodology for Structured Hypermedia Design. *Communications of the ACM* 38 (8), pp. 34-44.
- LYARDET F. D., ROSSI G. H., SCHWABE D., (1998) Using Design Patterns in Educational Multimedia applications, *ED-MEDIA '98*.
- MARQUEZ, J.M., (2002) *Eduka. Primer estudio de aproximación a un sistema de educación semipresencial*. Proyecto Fin de Carrera para la obtención del título de Ingeniero en Informática. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla.
- ORTEGA, J.A., TORRES J. Y MÁRQUEZ J.M., (2001) *Aplicación de las nuevas tecnologías: Docencia Interactiva*. Revista de Enseñanza Universitaria 2001, 18; pp 115-127.
- SCHWABE D., ROSSI G., Y BARBOSA S. D. J., (1996). Systematic Hypermedia Application Design with OOHDM. *Hypertext'96 (HT96)*, Washington.
- ROSSI G. Y SCHWABE D. *The Object-Oriented Hypermedia Design Model*. <http://www-lifia.info.unlp.edu.ar/~fer/oohdm/>
- WALL STREET INSTITUTE. *The Village, the Wall Street Institute Student Community*. Extranet mundial: <http://www.e-wsi.com>

MULTIÁREAS