

Propuesta de un modelo navegacional para el desarrollo de aplicaciones basadas en OOHDM

Ricardo Soto
rsoto@inf.ucv.cl

Silvana Roncagliolo
silvana@ucv.cl

Wenceslao Palma
wenceslao.palma@ucv.cl

Escuela de Ingeniería Informática
Universidad Católica de Valparaíso
Av. Brasil 2241 – Casilla 4059
Fono 32 273761- Fax 32 273859
Valparaíso, Chile

Resumen

En la actualidad son pocas las metodologías existentes que permiten a los desarrolladores conseguir productos de software hipermedia reusables y fáciles de mantener. A pesar de ello, ha nacido una tendencia a considerar el desarrollo hipermedial con un enfoque de proceso de ingeniería (de software), por lo que ya se han propuesto algunas metodologías para este fin. Una de ellas es OOHDM (Object Oriented Hypermedia Design Method), la cual plantea un proceso de cinco fases en el que se combinan diferentes estilos de desarrollo. El principal objetivo de este trabajo es presentar un prototipo de modelo navegacional, capaz de evitar redundancia encontrada en los diagramas presentados por dicha metodología.

***Palabras Claves:** Hipertexto, Multimedia, Hipermedia.*

1. Introducción

El gran avance de las comunicaciones y las tecnologías de información, acompañado de necesidades cada vez más exigentes en el mundo de la informática han incrementado exponencialmente el campo hipermedial. Nuevas áreas de aplicación, que necesitan rápidamente la flexibilidad del hipertexto unida a una rica variedad de datos multimedia, han nacido durante los últimos cinco años. Lamentablemente, la construcción de grandes aplicaciones hipermedia es extremadamente difícil, por otro lado no existe una metodología que se adapte perfectamente a este tipo de software, tentando a los desarrolladores a la omisión del diseño estructural de la aplicación. Esta falencia difícil de remediar generalmente entrega como resultado un software de baja calidad y susceptible de correcciones posteriores. Sin exagerar, la etapa de mantención sigue siendo un problema, el no contar con la documentación adecuada de la aplicación, significa transformar el proceso de mantención en una tarea agobiante.

El comienzo de la solución a estos problemas nace principalmente en la creación de una adecuada programación de tareas antes de la construcción de la aplicación, para lograr esto surge la necesidad de definir metodologías de desarrollo que utilicen modelos y estructuras formales de diseño e implementación, especialmente orientadas a software hipermedia.

Habitualmente el desarrollo de Sistemas Hipermediales suele hacerse utilizando directamente herramientas a nivel de implementación, descuidándose el importante proceso previo de análisis y diseño de los aspectos estructurales de la navegación e interfaz. Sin embargo, en los últimos años existe una tendencia a considerar el desarrollo hipermedial con un enfoque de proceso de ingeniería (de software), por lo que ya se han propuesto diferentes metodologías, como HDM (Hypertext Design Model) [1], EORM (Enhanced Object Relationship Model) [2], RMM (Relationship Management Methodology) [3] u OOHDM (Object Oriented Hypermedia Design Method) [4], que consideran 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 hipermedial a desarrollar.

El documento está organizado de la siguiente manera: la sección 2 explica brevemente la metodología utilizando un caso práctico, la sección 3 describe las ventajas y desventajas de la metodología, la sección 4 presenta

una propuesta de modelo navegacional como posible mejora a una de las principales debilidades de OOHDM y la sección 5 conclusiones y trabajo futuro.

2. Un vistazo a OOHDM

OOHDM es una metodología orientada a objetos que propone un proceso de desarrollo de cinco fases donde se combinan notaciones gráficas UML con otras propias de la metodología. En una primera instancia debido al poco auge que tenía Internet, OOHDM era sólo para aplicaciones que incluían hipertexto y algo de multimedia (CD-ROM promocionales, enciclopedias, museos virtuales, etc.). Pero el gran desarrollo de Internet obligó su adaptación para el desarrollo de aplicaciones hipermedia en Internet, tales como comercio electrónico, motores de búsqueda, sitios educativos y de entretenimiento. En la figura 1 se muestran las cinco etapas de OOHDM.

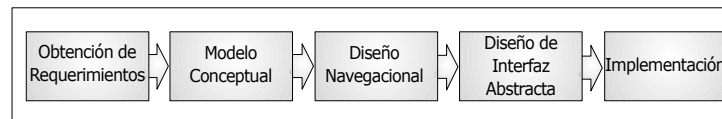


Figura 1 Las cinco etapas de la metodología OOHDM.

2.1. Ejemplo práctico

A continuación un caso con el que se ejemplificarán las cinco etapas de la metodología:

“All Horizons” es una empresa que ofrece servicios de capacitación a distintas empresas a nivel nacional. Su principal fuerte son los cursos y seminarios relacionados con temas informáticos. La idea es desarrollar un sitio “web” que sea capaz de ofrecer información en forma intuitiva de los cursos y seminarios que se imparten. Además sería óptimo agregarle pequeñas funcionalidades, tales como, permitir a los usuarios bajar los textos y documentos relacionados con el curso que han tomado o darles la posibilidad de ver su nota obtenida en el curso.

2.2. Obtención de requerimientos

Como en todo proyecto informático la obtención de requerimientos es una de las etapas más importantes, la mayoría de los estudios entregan resultados claros que los errores más caros son los que se cometen en esta etapa. Para enfrentar esta dificultad, OOHDM propone dividir esta etapa en cinco subetapas: Identificación de roles y tareas, Especificación de escenarios, Especificación de casos de uso, Especificación de UIs (User Interaction Diagram) y Validación de casos de uso y UIs [5].

2.2.1. Identificación de roles y tareas

En esta subetapa el analista deberá introducirse cuidadosamente en el dominio del sistema, ahora su principal labor será identificar los diferentes roles que podrían cumplir cada uno de los potenciales usuarios de la aplicación.

Los usuarios juegan roles importantes en cada intercambio de información con el sistema. En el ejemplo, una examinación inicial podría revelar los siguientes posibles roles: Alumno, Potencial Alumno, Profesor, Agente de Ventas, Secretaria, Coordinador. Para efectos de validación de los casos de uso es muy importante tener identificado el rol de cada usuario, ya que serán ellos los que entregarán su conformidad con respecto al caso de uso en el que participan.

Luego para cada rol el analista deberá identificar las tareas que deberá soportar la aplicación, como por ejemplo para el rol estudiante: Buscar información acerca de un curso, Buscar información acerca de un profesor u Obtener el material para un curso.

2.2.2. Especificación de escenarios

Los escenarios son descripciones narrativas de cómo la aplicación será utilizada [6]. En esta subetapa, cada usuario deberá especificar textual o verbalmente los escenarios que describen su tarea. En la figura 2 se muestran dos escenarios obtenidos en el ejemplo.

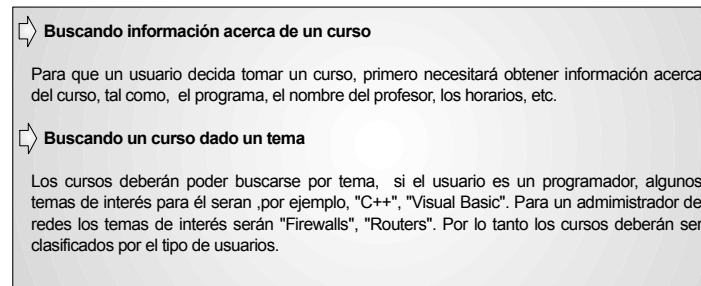


Figura 2 Escenarios especificados por usuarios en el caso de estudio.

2.2.3. Especificación de casos de uso

Un caso de uso es una forma de utilizar la aplicación. Específicamente representa la interacción entre el usuario y el sistema, agrupando las tareas representadas en los escenarios existentes. Es muy importante que el analista identifique cuál es la información relevante en cada uno de ellos, para luego generar un caso de uso coherente. En figura 3 se muestra el caso de uso "Buscando un curso dado un tema".

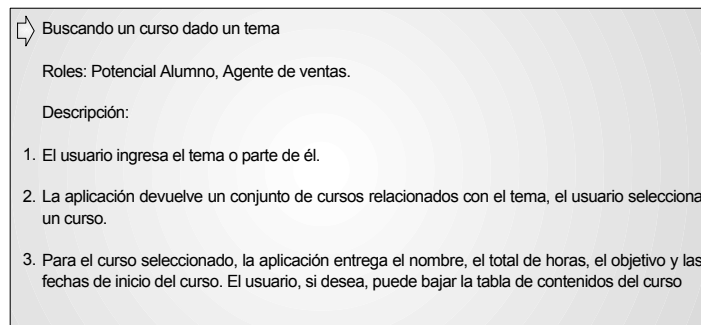


Figura 3 Caso de uso "Buscando un curso dado un tema".

2.2.4. Especificación de UIDs

De acuerdo a UML, los diagramas de secuencia, de colaboración y de estado son capaces de representar un caso de uso. Sin embargo, la especificación de casos de usos usando estas técnicas es un amplio trabajo y puede anticiparse inesperadamente a tomar algunas decisiones de diseño [5]. Para evitar esto OOHDM propone la utilización de una herramienta, llamada UID, que permite representar en forma rápida y sencilla los casos de uso generados en la etapa anterior.

Para obtener un UID desde un caso de uso, la secuencia de información intercambiada entre el usuario y el sistema debe ser identificada y organizada en las interacciones. Identificar la información de intercambio es crucial ya que es la base para la definición de los UIDs. En la figura 4 se muestra el UID relativo al caso de uso "Buscando un curso dado un tema".

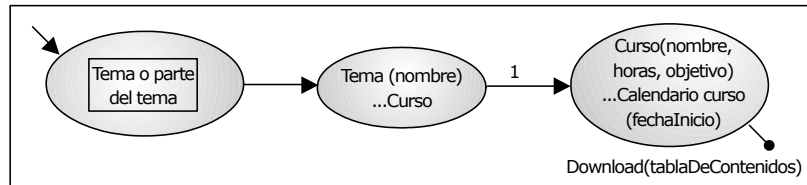


Figura 4 UID correspondiente al caso de uso “Buscando un curso dado un tema”.

2.2.5. Validación de casos de uso y UIDs

En esta etapa, el desarrollador deberá interactuar con cada usuario para validar los casos de uso y UIDs obtenidos, mostrando y explicando cada uno de ellos para ver si el o los usuarios están de acuerdo. El usuario deberá interceder sólo en aquellos casos de uso y UIDs en que participa.

2.3. Diseño conceptual

En esta etapa se genera un modelo conceptual, donde las clases, relaciones y cardinalidades se definen de acuerdo a reglas que se aplican sobre los UIDs. Cabe destacar que gran parte de ellas provienen de las técnicas de normalización [5]. En la figura 5 se muestra el esquema conceptual resultante.

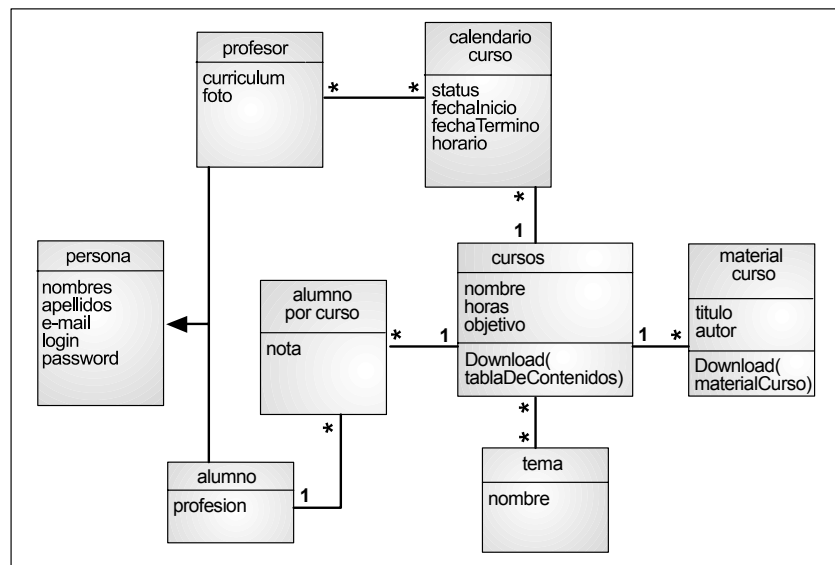


Figura 5 Esquema conceptual resultante.

2.4. Diseño navegacional

En esta etapa de la metodología se pretende desarrollar una topología navegacional que permita a la aplicación ejecutar todas las tareas requeridas por el usuario. La idea principal es unificar una serie de tareas para obtener el diseño navegacional de la aplicación [7].

Para cada UID se crearán diagramas de contexto y tarjetas de especificación que detallan la información contenida en el diagrama. En la figura 6 se muestra el diagrama de contexto correspondiente al UID del caso de uso “Buscando un curso dado un tema”.

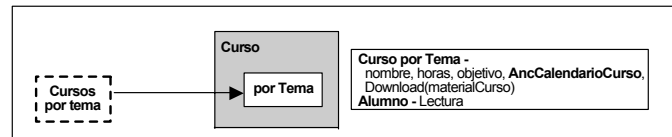


Figura6 Diagrama de contexto correspondiente al UID del caso de uso “Buscando un curso dado un tema”.

2.4.1. Aplicación del diseño navegacional.

Una vez que ya se han diseñado todos los diagramas de contexto, uno para cada caso de uso con sus respectivas tarjetas de especificación, es necesario realizar la unión de todos los diagramas para formar uno sólo. El diagrama resultante corresponderá al diagrama de contexto de toda la aplicación. La figura 7 muestra el diagrama resultante de la unión de todos los diagramas de contexto obtenidos.

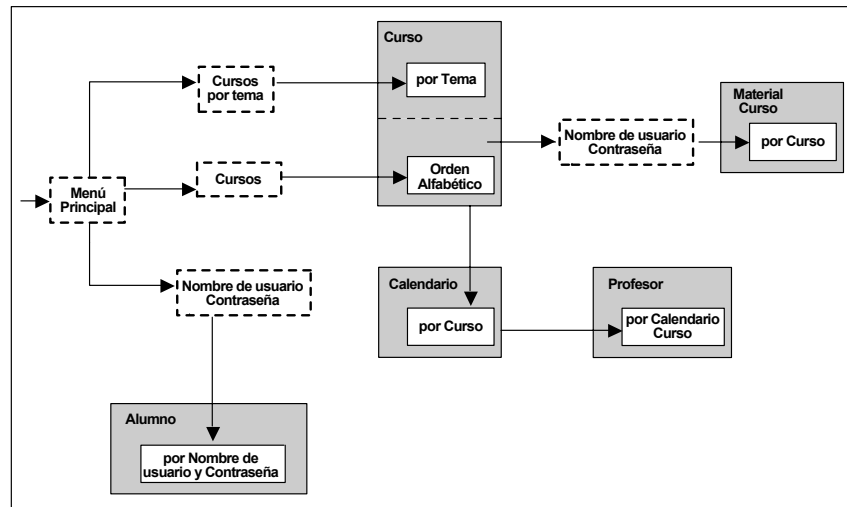


Figura 7 Diagrama de contexto final.

2.4.2. Esquema de clases navegacionales.

El diseño navegacional en OOHDH corresponde a un conjunto de modelos que se van desarrollando paso a paso, ya se ha desarrollado el diagrama de contexto con sus respectivas tarjetas de especificación. En la siguiente tarea corresponde desarrollar el esquema de clases navegacionales [7], este modelo corresponde a una combinación entre el modelo conceptual y el diagrama de contexto, donde las clases navegacionales son llamadas nodos, las relaciones navegacionales se llaman vínculos y los atributos de los nodos que activan navegaciones son llamados anclas.

2.5. Diseño de interfaz abstracta

Una vez finalizado el diseño navegacional, será necesario especificar las diferentes interfaces de la aplicación. Esto significa definir de que manera aparecerán los objetos navegacionales en la interfaz y cuales objetos activarán la navegación. Para lograr esto se utilizarán ADVs (Vista de Datos Abstracta) [4,8], modelos abstractos que especifican la organización y el comportamiento de la interfaz, es necesario aclarar que las ADVs representan estados o interfaces y no la implementación propiamente tal. La figura 8 muestra la ADV de “Curso por tema”.

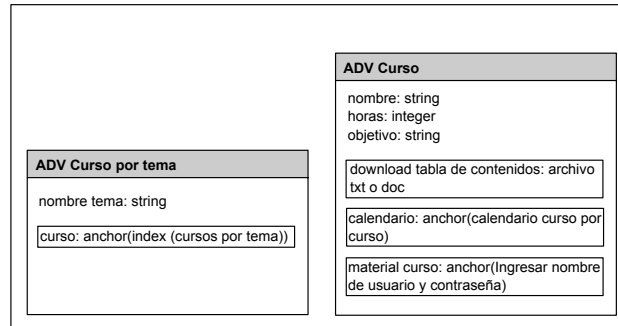


Figura 8 ADVs relacionadas con el caso de uso “Buscando un curso dado un tema”.

2.6. Implementación

Una vez terminadas las etapas anteriores, el desarrollador posee un completo conocimiento del dominio del problema. Así entonces, ya ha identificado la información que será mostrada, como estará organizada y cuales funciones permitirá ejecutar la aplicación. Además de ello, cuenta con una idea básica de cómo se verán las interfaces.

Para comenzar con la implementación el desarrollador deberá elegir donde almacenará los objetos y con qué lenguaje o herramienta desarrollará las interfaces, es necesario aclarar que generalmente el desarrollador se encarga del lado técnico de la interfaz, la parte gráfica y el que le dará la apariencia final a la interfaz será el diseñador gráfico.

3. Ventajas y desventajas de OOHDM

Ventajas

- OOHDM posee una notación diagramática bastante completa, que permite representar en forma precisa elementos propios de las aplicaciones hipertexto, tales como nodos, anclas, vínculos, imágenes, estructuras de acceso y contextos.
- En cada etapa de la metodología, especialmente en las de análisis y diseño, el usuario es considerado un integrante fundamental en la validación del producto obtenido. Esta interacción ayuda al desarrollador a entender y lograr en cada etapa lo que el usuario realmente necesita.
- OOHDM genera una cantidad considerable de documentación a través de sus distintas etapas de desarrollo, lo que permite llevar un control del desarrollo de las etapas y tener la posibilidad real de realizar una rápida detección, corrección de errores y mantención.
- OOHDM ofrece la posibilidad de crear estructuras de reuso, tales como los “esqueletos” o “frameworks”, cuyo principal objetivo es simplificar las tareas de diseño y disminuir su consumo de recursos [9,10,11].
- OOHDM utiliza una herramienta diagramática llamada UID, la cual es muy útil y sencilla de usar. Este instrumento es capaz de representar en forma precisa y con claridad los casos de uso obtenidos.

Desventajas

- Si bien es cierto los creadores de OOHDM señalan que la metodología fue creada principalmente para desarrollar aplicaciones hipertextuales de gran extensión. Dicha orientación ha llevado a los creadores a desarrollar una serie de reglas y pasos (a veces bastante complicados de seguir) para realizar distintos mapeos entre un diagrama y

otro, con el principal objetivo de simplificar y mecanizar las tareas de cada fase, este intento de mecanización puede traer como consecuencia el olvido de detalles fundamentales por parte del desarrollador.

- El diseño navegacional es un tanto tedioso, para resolverlo adecuadamente es necesario realizar una gran cantidad de diagramas que muchas veces entregan información similar a la entregada por los UIDs y las ADVs. Esta redundancia de información podría ser evitada graficando la información en un solo tipo de diagrama que sea capaz de reunir las capacidades de los UIDs, diagramas de contexto y ADVs.

4. Propuesta para un modelo navegacional.

Una de las desventajas importantes que posee OOHDM corresponde a la redundancia de información presentada por los diagramas necesarios de cada etapa, esta debilidad se puede demostrar revisando los diagramas para el caso de uso “Buscando un curso dado un tema” correspondientes a las figuras 4, 6 y 8. De estas figuras es sencillo notar que existe información repetida. Por ejemplo la tarjeta de especificación “Curso por tema” del diagrama de contexto posee prácticamente la misma información que la tercera interacción del UID y la ADV “Curso”. Además la secuencia navegacional se puede deducir tanto desde el UID como del diagrama de contexto.

Debido a que este problema no es sólo una coincidencia y se produce en reiteradas ocasiones, este trabajo tiene como principal objetivo presentar un prototipo de estructura navegacional, capaz de evitar esta redundancia encontrada en los diagramas presentados por OOHDM.

La idea planteada se implementa graficando los nodos, no de manera abstracta como se realiza en una ADV, sino de manera real, es decir, lo más cercano posible a lo que se pretende que aparezcan cuando estén implementados. Una vez que se han graficado los nodos se procede a unir cada uno de ellos para demostrar su navegación, de una manera bastante similar a como se realiza en la etapa de diseño navegacional de OOHDM.

4.1 Graficación de nodos.

A continuación se utilizará el prototipo de estructura navegacional propuesto para desarrollar el diseño navegacional, sobre la base del sitio “All Horizons” presentado anteriormente. En la figura 9 se muestra la gráfica del menú principal, el nodo posee tres “frames” o “marcos” uno superior fijo que muestra una imagen (podría ser una imagen con el logo de la empresa), uno lateral izquierdo, también fijo donde están las estructuras de acceso “Cursos por tema”, “Cursos”, “Nombre de usuario” y “Contraseña” y un frame principal activo, este frame es el único que va cambiando su apariencia y es donde se presenta la información que va requiriendo el usuario. En este nodo hay dos tipos de estructuras de acceso, “Cursos” es un botón, “Nombre de usuario”, “Contraseña” y “Cursos por tema”, corresponden a cuadros de texto, la notación gráfica en la figura 9 los distingue.

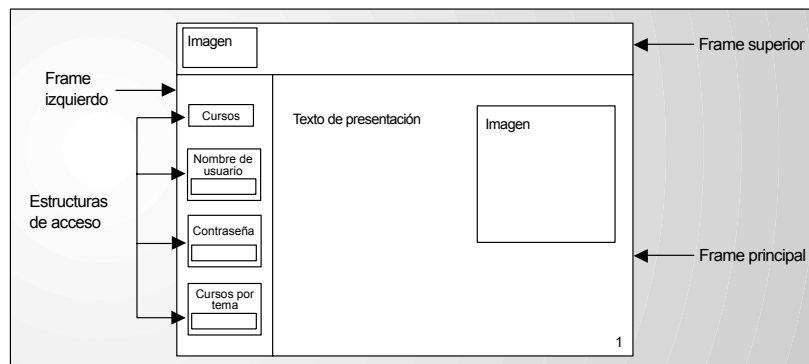


Figura 9 Diagrama del nodo que representa al menú principal de la aplicación.

La figura 10 muestra el nodo “Cursos por tema”. En el frame principal se puede observar que está el nombre del tema y abajo una lista (1...N, indica que es una lista) donde aparece sólo el nombre de cada curso, los cursos que aparecen son los que están relacionados al tema que ingresó el usuario. Cada curso está subrayado, lo que indica que

cada curso que aparece posee un hipervínculo a otro nodo, en este caso particular el hipervínculo es a un nodo llamado “Curso”, donde aparece información específica de cada curso. En la figura 11 se muestra el nodo “lista de todos los cursos”. En este nodo aparece una lista de cursos en orden alfabético, para cada curso sólo se muestra el nombre, igual que en el caso anterior están subrayados, lo que significa que existe un hipervínculo a otro nodo. En la figura 12 se muestra el nodo “Curso”, en este nodo aparece el nombre, las horas y el objetivo del curso, además aparece una lista con todas las fechas de inicio que posee este curso, cada fecha de inicio tiene un hipervínculo a otro nodo, en el cual se dará más información con respecto a la impartición de ese curso para esa determinada fecha de inicio. Más abajo hay un hipervínculo hacia el nodo “Material del curso” y finalmente se presenta la posibilidad de bajar la tabla de contenidos del curso.

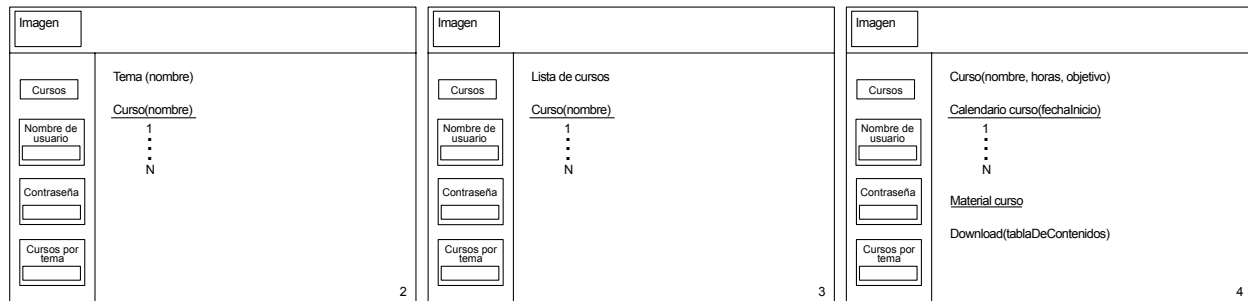


Figura 10, 11 y 12 Diagramas de los nodos “Cursos por tema”, “Lista de todos los cursos”, “Lista de todos los cursos” respectivamente.

En la figura 13 se muestra el nodo “Calendario del curso”, en este nodo se visualiza el nombre del curso, los datos referentes a la impartición de un curso en una determinada fecha (status, fecha de inicio, fecha de término y horario) y una lista de los profesores que dictan ese determinado curso para esa determinada fecha. Para cada profesor sólo se muestran los nombres y los apellidos, información en detalle se podrá obtener por medio del hipervínculo que poseen. En la figura 14 se muestra el nodo “Profesor”, donde aparece en detalle la información de un profesor. En la figura 15 se muestra el nodo “Notas de un alumno”, donde aparecen los nombres y apellidos del alumno. Además se muestra la nota con el respectivo nombre del curso al lado, para todas las asignaturas que el alumno a cursado.

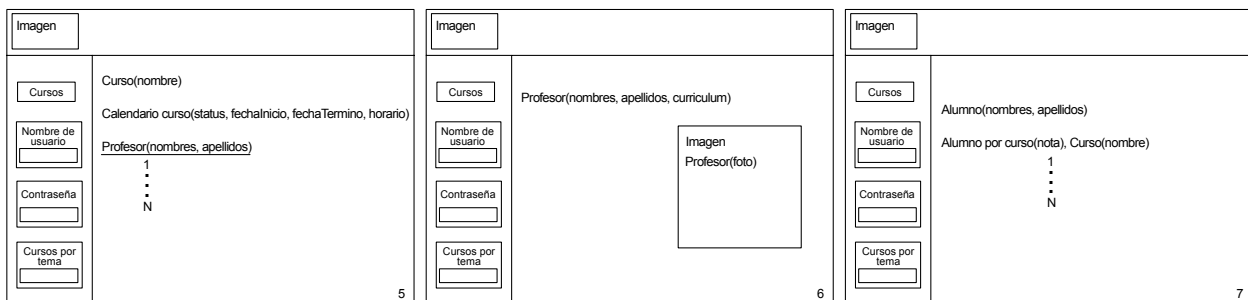
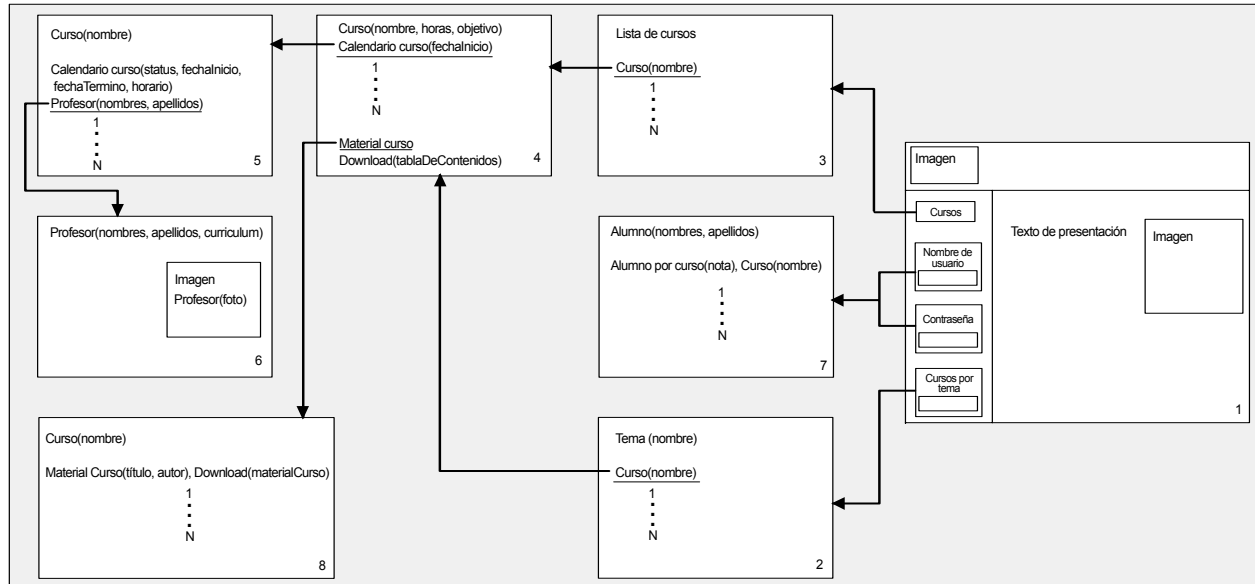


Figura 13, 14 y 15 Diagramas de los nodos “Calendario del curso”, “Profesor”, “Notas del alumno” respectivamente.

Una vez finalizados los nodos se recomienda realizar tarjetas de especificación para las consultas y para los requerimientos que no se hayan podido representar gráficamente. Las tarjetas de especificación pueden tener un formato similar a las tarjetas presentadas en la etapa de diseño navegacional de OOHDM.

4.2. Modelo navegacional

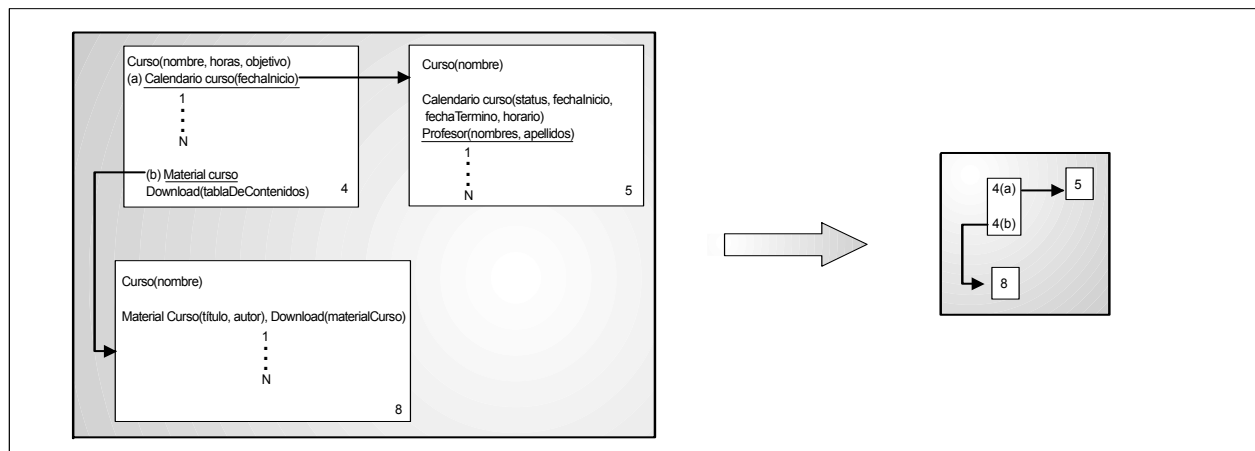
En esta etapa se procederá a unir cada nodo obtenido para realizar el modelo navegacional. La figura 16 muestra el modelo navegacional de la aplicación.



Figuras 16 Modelo navegacional.

Para los nodos, a excepción del nodo “menú principal”, se visualiza sólo el frame principal, para simplificar la visión del modelo. No existe implicancia alguna ya que los frames que no aparecen no cambian durante la navegación.

La navegación se deduce del diagrama, desde el menú principal se puede acceder a los nodos “Cursos por tema”, “Lista de todos los cursos” y “Notas de un alumno”. Desde los nodos “Cursos por tema” y “Lista de todos los cursos” se puede obtener el nodo “Curso”, desde éste se puede llegar al nodo “Material del curso” y al nodo “Calendario del curso” y de éste al nodo “Profesor”. Además, cada nodo posee un número de identificación ubicado en la esquina inferior derecha, este número se puede utilizar cuando existan muchos nodos y el modelo navegacional sea demasiado grande, en este caso sólo se hace referencia al número del nodo. Cuando el nodo posea más de un hipervínculo también será necesario identificarlo, para saber exactamente desde cual hipervínculo se inició la navegación. La figura 17 muestra la identificación de los hipervínculos “Calendario curso(fechaInicio)” y “Material curso”.



Figuras 17 Identificación de los hipervínculos y nodos.

Para finalizar la propuesta es necesario mostrar como quedarían las etapas de la metodología con la incorporación del nuevo modelo navegacional. En una primera instancia se plantea reemplazar sólo los diagramas de contexto y ADVs, ya que los UIDs son muy útiles para la etapa de obtención de requerimientos y para la creación del

modelo conceptual. No obstante cuando el desarrollador haya adquirido mayor experiencia, sería posible reemplazar los 3 diagramas de OOHDM por el modelo navegacional propuesto. En la figura 18 se muestran las etapas de la metodología modificada.

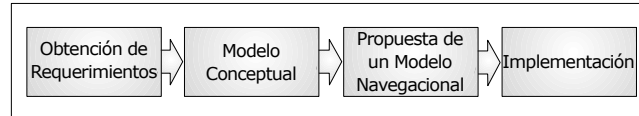


Figura 18 Las etapas de la metodología OOHDM con la incorporación del nuevo modelo navegacional.

5. Conclusiones y trabajo futuro

El campo hipertextual e Internet crecen todos los días, la sociedad requiere cada vez sistemas más complejos y más grandes; y el gran problema radica en la escasa cantidad de recursos y herramientas que se dispone para enfrentar este rápido crecimiento. Por ello es de gran urgencia para el área informática, comenzar a utilizar en forma estandarizada una metodología de desarrollo que ayude a crear aplicaciones fáciles de mantener y posibles de reusar.

La incorporación de nuevas metodologías y modelos que aporten al desarrollo de aplicaciones hipertexto es de gran importancia para el desarrollo de esta área.

El modelo navegacional propuesto en este trabajo es el inicio de un completo estudio del proceso de desarrollo de este tipo de software, el cual será aplicado en el desarrollo de distintos tipos de aplicaciones hipertexto con el objetivo de obtener experiencia para desarrollar en un futuro cercano una herramienta CASE que permita simplificar aún más todo el proceso de desarrollo de este tipo de software.

6. Referencias

- [1] Franca Garzotto, Paolo Paolini, Daniel Schwabe "HDM - A Model-Based Approach to Hypertext Application Design" ACM Transaction on Information Systems, vol. 11, nº 1, January 1993, pages 1-26.
- [2] D.B. Lange, "An Object-Oriented Design Approach for Developing Hypermedia Information Systems", Research Report RT0112, IBM Research, Tokyo Research Laboratory, Japón, 1995.
- [3] Tomas Isakowitz, Edward A. Stoh, P. Balasubramanian, "RMM: A Methodology for Structured Hypermedia Design. Communication of the ACM, August 1995.
- [4] Schwabe, D., and Rossi, G. An object-oriented approach to Web-based application design. Theory and Practice of Object Systems (TAPOS) (October 1998), 207-225.
- [5] Vilain, P., Schwabe, D., de Souza, C. S.: A Diagrammatic Tool for Representing User Interaction in UML. To appear in UML 2000 -Third International Conference on the Unified Modeling Language, (York, UK, October, 2000).
- [6] Vilain, P., Schwabe, D., de Souza, C. S.: Use Cases and Scenarios in the Conceptual Design of Web Applications. Technical Report MCC 12/00, Departamento de Informática, PUC-Rio (2000)
- [7] "Modeling Interactions and Navigation in Web Applications", Lecture Notes in Computer Science 1921, Proceedings of the World Wide Web and Conceptual Modeling'00 Workshop, ER'00 Conference, Springer, Salt Lake City, 2000. (Extended version)
- [8] D. Schwabe, G. Rossi "Developing Hypermedia Applications using OOHDM"(Rio, Brazil, 1998.)
- [9] D. Schwabe, G. Rossi, L. Esmeraldo, F. Lyardet: "Web Design Frameworks:An approach to improve reuse in Web applications. Proceedings of the WWW9 Web Engineering Workshop, Springer Verlag LNCS, forthcoming.
- [10] D. Schwabe, G. Rossi, L. Esmeraldo, F. Lyardet: "Engineering Web Applications for reuse". To appear, IEEE Multimedia, Spring 2001.
- [11] Rossi, G., Schwabe, D., and Lyardet, F. Improving Web information systems with navigational patterns, inProceedings of 8th International World Wide Web Conference (Toronto, Canada, May 1999), Elsevier Science, 589-600.
- [12] Rossi, G., Schwabe, D., Lyardet, F.: Web application models are more than conceptual models. In: Proceedings of the World Wide Web and Conceptual Modeling'99 Workshop, ER'99 Conference. Lecture Notes in Computer Science. Springer (1999) 239-252.