

COMPARACIÓN DE MODELOS DE DISEÑO DE HIPERMEDIA

Cernuzzi Luca, Stecher Rodolfo

Departamento de Ingeniería Informática
Universidad Católica "Nuestra Señora de la Asunción"
Fax: +595 - 21 - 310.587, Asunción - Paraguay
Email: lcernuzz@uca.edu.py

Resumen

El presente trabajo enfoca un estudio de diferentes modelos y metodologías para el diseño de aplicaciones hipermediales. Se presenta una comparación en términos de ventajas y desventajas de la utilización de algunos de los principales modelos y metodologías seleccionados en la especificación de cierto tipo de aplicación hipermedial.

Palabras Claves: Hipermedia, Metodologías Hipermedia.

1. INTRODUCCIÓN

Con una creciente velocidad, la tecnología hipermedia se integra a áreas ya existentes de la computación por la capacidad integradora que tiene, puesto que relaciona y permite visualizar y recorrer varios tipos de información a la vez, logrando así una mejor comprensión por parte del usuario. Debido a esta capacidad de enlace y de navegación a través de la información, las aplicaciones se convierten en complejas estructuras no lineales. La cantidad de información disponible crece, y con ello el problema de manejar su diseño y mantenimiento [Isako95, Bieber97] como también se hace perentoria la reutilización de componentes y de soluciones ya desarrolladas debido al alto costo de hacer cada aplicación desde cero. Por esto surgen los modelos y las metodologías de diseño que intentan ordenar dicha actividad.

Tanto la característica de hipermedia de envolver varios aspectos distintos, como la navegación, la interfaz, el almacenamiento del contenido, etc., así como experiencias previas de construcción de hipermedia realizadas en la Universidad Católica "Nuestra Señora de la Asunción" nos indican que los modelos tradicionales de la Ingeniería de Software como Diagramas de Flujo de Datos, diagramas Entidades-Relaciones y la Orientación a Objetos no pueden por sí mismos representar las complejas características de este tipo de aplicaciones.

En pos de apuntar dicho problema, una variedad de modelos y metodologías de diseño [Stott89, Tompa89, Garzo91, Hardm93, Halas94, Lange94, Isako95, Rossi96] han sido propuestos desde final de la década pasada. También algunos escasos trabajos de evaluación de dichas metodologías han sido presentado [Cern976, Chris98]. Sin embargo, no se encuentran en la literatura especializada pautas o trabajos que permitan a los autores disponer de una guía que facilite la elección de los modelos o metodologías a utilizar en cada caso, dependiendo de las características (de estructura, contenido, variación del contenido en el tiempo, etc.) de la aplicación objetivo.

Este problema, que se presenta a los autores cada vez que quieren producir una nueva aplicación, motiva el presente estudio que pretende constituir un primer aporte en cuanto a la evaluación de modelos y metodologías.

2. DESCRIPCIÓN DEL PROBLEMA

La literatura especializada ofrece una amplia gama de modelos y metodologías para el diseño de aplicaciones y sistemas hipermedia: Dexter Hypertext Reference Model [Halas94], Amsterdam Hypermedia Model [Hardm93, Hardm94], Trellis Model [Stott89], Hypergraph Data Model [Tomba89], "Hypertext Design Model" – HDM [Garzo91, Garzo93], "Enhanced Object Relationship Model" – EORM [Lange94], "Relationship Management Methodology" – RMM [Isako95] y "Object Oriented Hypertext Design Model" – OOHDM [Schwa 95, Rossi96].

Cada autor o grupo de trabajo manifiesta que el modelo o metodología presentado sirve o es útil para resolver algunos de los problemas conocidos del diseño de aplicaciones de este tipo, más el usuario interesado en diseñar una aplicación de cierto tipo y estructura no tiene una manera de saber cual sería el modelo o metodología más apropiado para la tarea a realizar. Dada la amplitud de aplicabilidad de este tipo de aplicaciones y la variedad de estructuras y conformaciones que pueden llegar a tener, no todos los modelos o metodologías son útiles en el mismo grado para resolver los problemas que se pudieran presentar.

Por este motivo, el principal objetivo al que se aboca el presente trabajo es dar un aporte en una análisis comparativo de distintos modelos y metodologías, de entre los propuestos, verificando si efectivamente ayudan y en que medida a solucionar los problemas que motivaron su aparición. Los autores consideran que la utilización de modelos y metodologías de diseño de hipermedios pueden ayudar al desarrollo de aplicaciones con cualidades deseables como ser una interfaz consistente, buenos mecanismos de navegación, una ordenada y adecuada estructuración de la información, etc. En la literatura es posible encontrar algunos ejemplos de criterios de calidad de la aplicación [Garzo95, Shnei92]. Consideramos que el grado de propiedades deseables, comunes a todas las aplicaciones desarrolladas con un modelo o metodología dado, nos brinda pautas de evaluación de este modelo o metodología y constituye una posible fuente (indirecta) de criterios de evaluación para modelos y metodologías de desarrollo hipermedial. Sin embargo, para realizar un trabajo comparativo serio es importante contar con un marco de evaluación global [Ortiz 99] que básicamente consta de un conjunto de criterios y un procedimiento para aplicar estos criterios. Los resultados obtenidos al aplicar este marco evaluativo no serán únicos ni definitivos, sin embargo es de esperarse que puedan orientar a una más adecuada aplicación de los modelos y metodologías de diseño de hipermedia y posiblemente proveer sugerencias para los autores.

En la perspectiva de evaluación formativa que nos ocupa lo más importante es tratar de identificar fortalezas y debilidades de modelos y metodologías ya existentes con respecto a cierto tipo de aplicación. Este último aspecto es muy relevante, en efecto, adentro del área hipermedial, el campo aplicativo es tan amplio que difícilmente un único modelo o metodología pueda ser el más eficiente para todos los casos.

Es así que, en una primera etapa se elige el tipo de aplicación. En un segundo momento se selecciona el o los modelos y metodologías que se pretenden evaluar en base a un subconjunto de criterios del marco de evaluación [Ortiz 99].

2.1 Selección de la Aplicación

Para esta tarea se seleccionó la aplicación "Itaipú, El proyecto del Siglo" desarrollada por nuestro grupo en la Universidad Católica "Nuestra Señora de la Asunción" (Paraguay) pues encontramos similitudes entre esta aplicación y aplicaciones futuras que nos habíamos propuesto realizar.

Se trata de una aplicación informativa altamente estructurada y tener una escasa repetición de contenidos, presentando al proyecto de la hidroeléctrica Itaipú, mencionando los distintos aspectos y partes de la misma.

Las páginas de la aplicación, sean las de índice o las de contenidos, tienen una estructura uniforme a lo largo de la aplicación, estando cada tema organizado en una estructura o jerarquía descendente. Cada nivel de la jerarquía de componentes es accedido por medio de un índice. Al fondo de la jerarquía se acaban los índices y quedan solo los contenidos a los que acceder.

Desde cualquier punto de la aplicación es posible acceder a una serie de funciones que permiten la navegación entre nodos de información y entre medios de presentación de acuerdo al contexto, la búsqueda de información y herramientas para recuperar la orientación.

Según la clasificación de aplicaciones hipermedia propuesta en [Isako95], "Itaipú, El Proyecto del Siglo" podría caracterizarse como de Baja volatilidad de la información y de mediana Estructuración. Con esto no estamos significando que no tiene estructura sino que las estructuras no son compartidas por todos los elementos.

2.2 Selección de los Modelos y Metodologías

Los primeros modelos propuestos Trellis [Stott89], Hypergraph Data Model [Tomba89] y Dexter [Halas94] (que en realidad es un modelo de referencia para sistemas hipermedia), fueron en general grandes guías a los modelos que surgieron más adelante. Un problema que presentan Trellis y Hypergraph es que utilizan formalismos de relativamente bajo nivel para la especificación de las aplicaciones, lo que los hace difícilmente manejables y poco cercanos a las herramientas implementativas actuales. Además, al crecer la hiperbase crece enormemente la complejidad de la especificación utilizando Redes de Petri (Trellis) o Autómatas (Hypergraph). Notamos que, comparando con los primeros modelos, las herramientas actuales están más cerca de los lenguajes de programación usados, lo que contribuye a su mayor difusión y utilización.

Por esto centraremos nuestro trabajo en la presentación de algunos de los modelos y metodologías que en nuestra consideración se destacan en los 90's y tienen cierta similitud en el nivel de abstracción logrado, a ser HDM [Garzo91, Garzo93], EORM [Lange94], RMM [Isako95] y OOHDM [Rossi96].

Entre los criterios incluidos en el marco global de evaluación existe uno que trata de la capacidad de los modelos y metodologías de realizar modelados de aplicaciones ya construidas, esto es, modelado a posteriori. Es este criterio el que tomamos para la realizar un análisis comparativo de los modelos y metodologías arriba mencionados, pues nos permite verificar además otros criterios generales como expresividad y generalidad, la independencia de la implementación y la capacidad del modelo de diseño de capturar el dominio subyacente.

3. MODELADO A POSTERIORI

En esta sección se presenta el modelado a posteriori de la aplicación seleccionada, obteniendo distintos resultados para cada uno de los modelos o metodologías previstos.

3.1. "El Proyecto del Siglo" - HDM

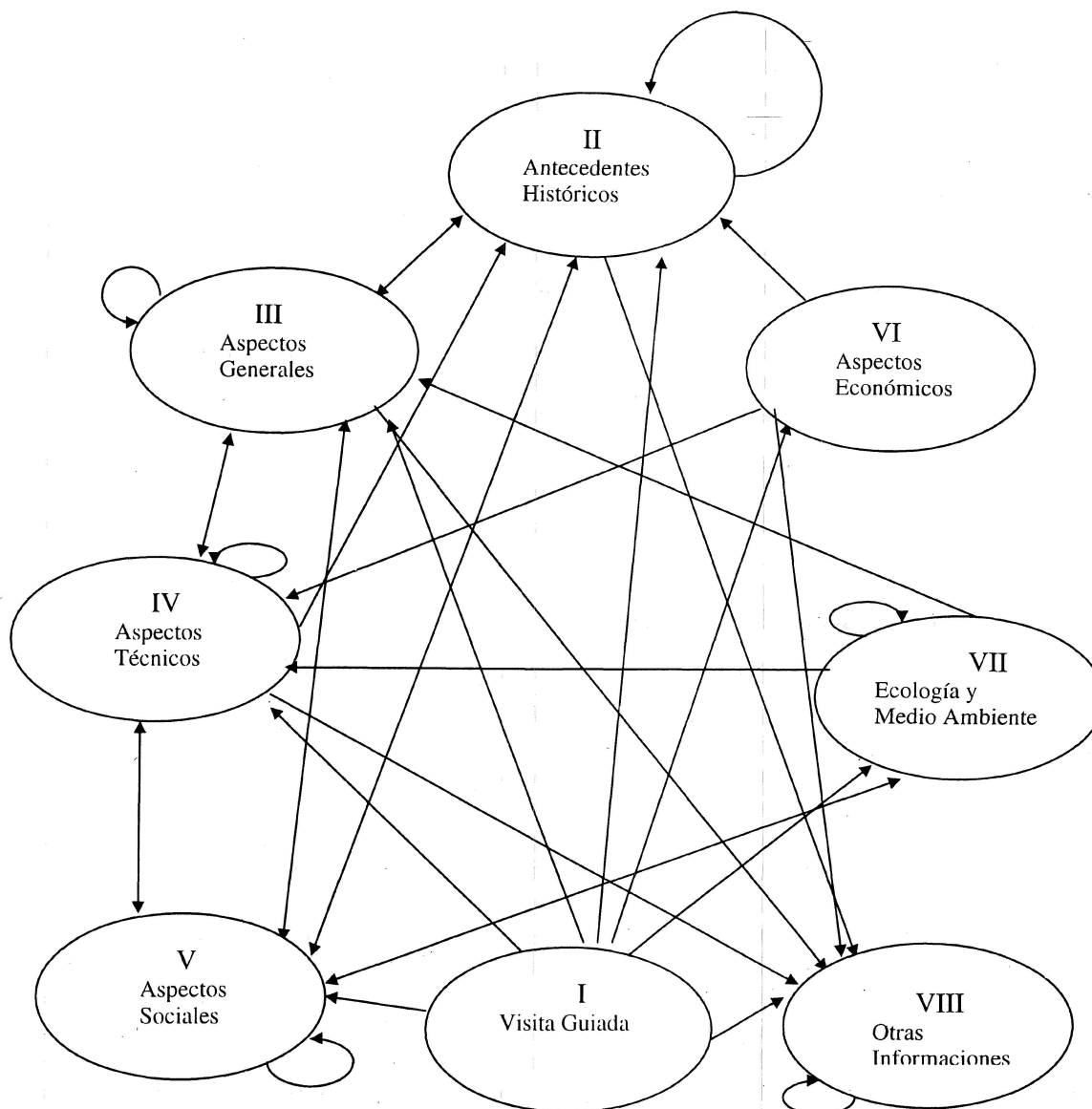


Fig. 1. Modelado del dominio en HDM

Para el modelado del dominio tenemos que al ser todas las entidades distintas, pues tienen distinta cantidad de componentes, no tenemos manera de generalizar estructuras.

Los enlaces aplicativos tampoco nos permiten reutilizar construcciones puesto que las entidades son distintas y por esto sería de poca utilidad definir tipos de enlaces en los que hay que definir tipo de entidad origen y destino en HDM. La noción de Perspectivas de HDM la utilizamos solamente entre el contenido textual y gráfico de las entidades, lo que a su vez implica componentes textuales y gráficos, unidades textuales y gráficas; sin embargo, ambas perspectivas son mostradas al mismo tiempo al usuario.

En lo que respecta al diseño de la navegación observamos que el modelo para representar la navegación en HDM se confunde con el modelo de representación del dominio, con lo que al final es difícil tener claro cuales eran elementos del dominio y cuales fueron agregados debido a requerimientos navegacionales. Tampoco podemos expresar los enlaces estructurales de forma adecuada, ya que tenemos enlaces entre componentes de una unidad y además enlaces entre visualizaciones del contenido de una unidad (que se visualiza en varios nodos). Otra desventaja es que con HDM difícilmente podemos representar la pertenencia de un conjunto de gráficos a cada componente y que cada componente se visualiza en distintos nodos (dependiendo de la longitud del texto). También un punto difícil de representar de manera general para poderlo utilizar donde haga falta son los textos con hotwords.

Esto nos induce a percatarnos de la poca independencia entre el modelado del dominio, el diseño navegacional y el diseño de la interfaz de HDM.

Para el diseño de la interfaz notamos que HDM no provee mecanismos para representarla en su aspecto estático ni mucho menos en su aspecto dinámico. HDM no provee primitivas para el modelado de la interfaz, con lo que nos vemos obligados a su descripción textual informal.

En el modelado de la aplicación resaltó evidente que su composición es heterogénea, es decir, que las estructuras no se repiten frecuentemente. Por este motivo, la representación de esta aplicación con HDM se torna larga y no se obtiene una de las ventajas importantes esperada al utilizar herramientas conceptuales para el diseño de una aplicación: la reutilización de estructuras similares especificándolas una sola vez.

Podríamos concluir que para este tipo de aplicación, el modelo HDM sirve de documentación extensiva de la aplicación, lo que constituye un aspecto interesante, más no es el principal objetivo de la utilización de un modelo.

3.2 “El proyecto del Siglo” - RMM

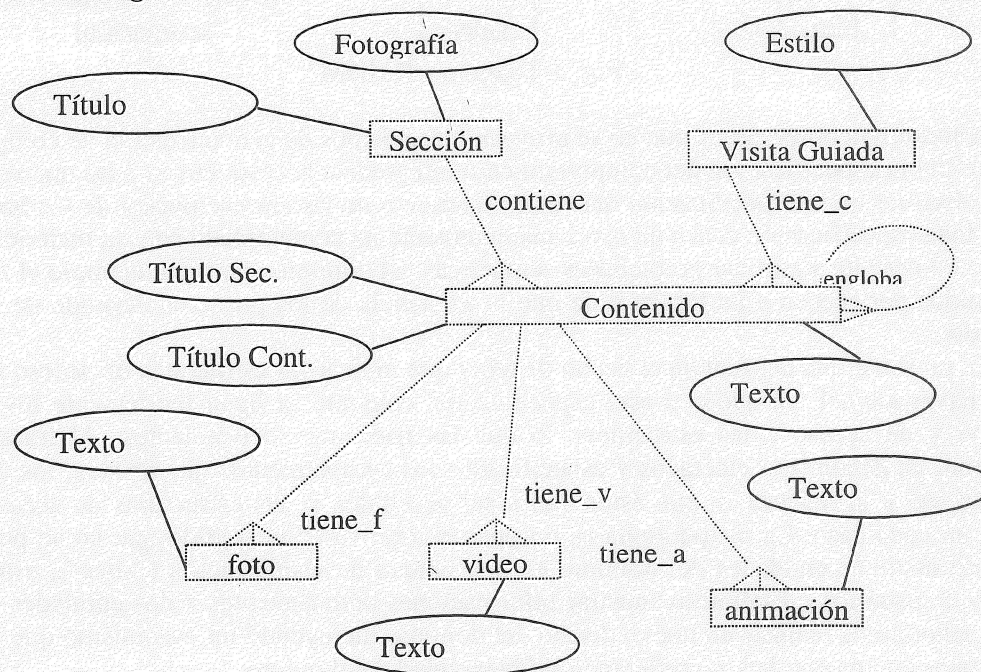


Fig. 2. Modelado del dominio en RMM

En este esquema puede apreciarse que el dominio de la aplicación fue representado de manera clara y concisa más con algunas falencias. La pertenencia de grupos de entidades a otras es difícil de representar, por ejemplo el caso del componente gráfico de un contenido que siempre existe y puede ser o una fotografía o un vídeo o una animación y también pueden ser varios de cada uno de ellos o, frecuentemente, algunos de ellos pueden no estar presente. Observamos también la repetición de relaciones a entidades, esto es, las relaciones que semánticamente son iguales deben repetirse varias veces en el diseño, una para cada tipo distinto de entidad envuelta, no pudiendo reutilizarlas. Además tampoco queda bien clara la estructuración del dominio.

A continuación presentamos el diagrama RMDM que incluye el diseño del dominio, con la separación en slices de las entidades y las estructuras navegacionales proveídas por el modelo.

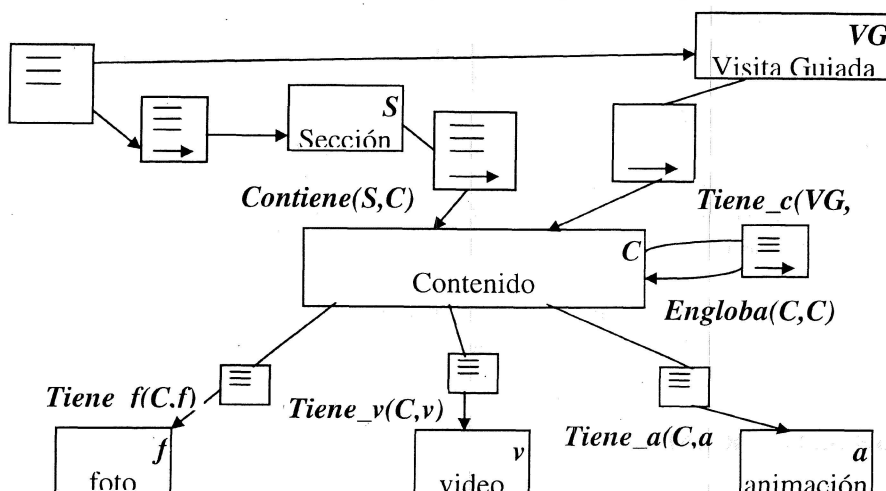


Fig. 3. Diagrama RMDM

En esta metodología observamos que no se proveen mecanismos de generalización de comportamientos ni se posibilita la especificación del comportamiento integrado a las estructuras a las que se aplica. Esto se puede observar en la especificación del comportamiento de las animaciones y de los botones que lo manejan. Esta especificación está a un nivel aparentemente no contemplado en esta metodología.

El modelo no permite significar propiedades semánticas del dominio como por ejemplo el hecho de que los contenidos pertenecen a un tópico y de que la existencia de los primeros depende de la existencia del segundo.

Asimismo, presenta una desventaja el hecho de tener que indicar las estructuras de acceso juntas con el esquema navegacional, ya que con esta especificación conjunta se ligan fuertemente los aspectos de navegación y de acceso a las estructuras, lo que las hace muy dependientes. Al tratar el aspecto navegacional ya deben ser detectados y especificados los agrupamientos que existen, los componentes de información y la manera en que éstos van a ser accedidos y, las estructuras de acceso que serán utilizadas en cada caso. La metodología es un poco restrictiva en el sentido que no se pueden incluir atributos de distintas entidades del dominio en una unidad de visualización ("slice"), esto es, si para cierto tipo de usuarios quisiéramos mostrar informaciones pertenecientes a dos entidades del dominio distintas, deberíamos realizar un nuevo diseño del dominio incluyendo un mecanismo que integre estas informaciones en una entidad, no reflejando así la realidad del dominio.

Finalmente, cabe destacar que RMM no provee herramientas conceptuales para el diseño de la interfaz.

3.3 “El Proyecto del Siglo” - EORM

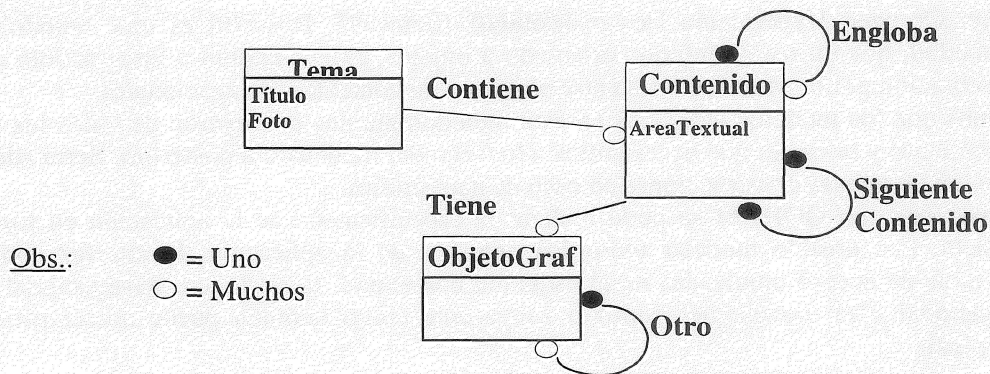


Fig. 4. Modelado del dominio en EORM

El marco de trabajo definido por el modelo nos lleva a esquemas que sirven como especificaciones de diseño, como documentación de las aplicaciones y, como ayuda y guía a la reutilización de diseños hipermediales. Este modelo, basado en el enfoque orientado a objetos de desarrollo de software, forma parte de un proceso de diseño en el que primeramente se procede a un análisis orientado a objetos utilizando un método de desarrollo orientado a objetos convencional como es el OMT (Object Modeling Technique), que captura la estructura, el comportamiento, y las interrelaciones de objetos en el dominio de la aplicación. El producto de esta fase permite producir el esquema EORM (descripción) del sistema que comprende la selección de entre varios tipos de enlaces para la representación de relaciones entre objetos, la construcción de estructuras compuestas y la definición de varios tipos de nodos. De este modo, la representación del dominio se hace en forma compacta e incluye su semántica. En la representación del dominio podemos apreciar los objetos del dominio y los relacionamientos entre ellos, por ejemplo el objeto **Tema**, que contiene al objeto **Contenido** y tiene atributos Título y Foto. O el objeto **Contenido**, que tiene **ObjetoGraf**, y tiene como atributo a **AreaTextual**. **ObjetoGraf** es una generalización de varios objetos gráficos y tiene estados y comportamiento definidos en el objeto. **AreaTextual** es también una generalización de objetos de tipo texto, cada uno con propiedades y comportamiento propio.

Teniendo en cuenta la interfaz, el acceso se realiza en base a índices, los que pueden contener otros índices. Todos los índices se presentan en una Página de Índice (aunque existe diferencia entre la página de Índice Principal y las otras). Los Contenidos, son como las hojas de las ramas creadas por los árboles-índices y presentan la información en sí, se presentan todos en Páginas de contenido similares, excepto los presentados en “Otras Informaciones” que tienen una estructura particular.

El modelo provee una serie de clases predefinidas de enlaces y relaciones entre objetos, como son los enlaces uni- y bi-direccionales y las relaciones de las jerarquías de generalización [Lange94]. Si el autor precisa utilizar otro tipo de enlaces o relaciones los puede definir construyéndolos desde cero o redefiniendo o aumentando los existentes.

Puede observarse que el diseño del dominio se confunde con el navegacional. Además, EORM no provee mecanismos para especificar los distintos contextos en que se visualizará y accederá a la información, debiendo el diseñador realizar un diseño navegacional para cada contexto distinto, y aún así no se obtiene la especificación deseada.

Finalmente, así como los modelos y metodologías anteriores, a este modelo le falta la provisión de mecanismos de representación de la interfaz.

3.4 “El Proyecto del Siglo” - OOHDM

“Object Oriented Hypermedia Design Method” [Schwa95, Rossi96] es una metodología de diseño hipermedial, que utiliza el enfoque orientado a objetos, extendiéndolo e integrándolo con técnicas de representación gráfica de relaciones entre objetos y de contextos navegacionales.

Notamos que los modelos utilizados en esta metodología nos proveyeron de todos los mecanismos de agrupamiento y herencia que necesitamos a lo largo del modelado a posteriori. Sería además factible la definición de nuevas construcciones en caso de necesitarlas.

Notamos que con OOHDM se pudo realizar la especificación de la aplicación en forma completa y compacta. Fue posible modelar todos los aspectos de la aplicación, la interfaz con los ADV, las estructuras de acceso empleadas en el esquema contextual, la estructura navegacional en el esquema navegacional y el modelo del dominio subyacente como también puede apreciarse en el esquema conceptual.

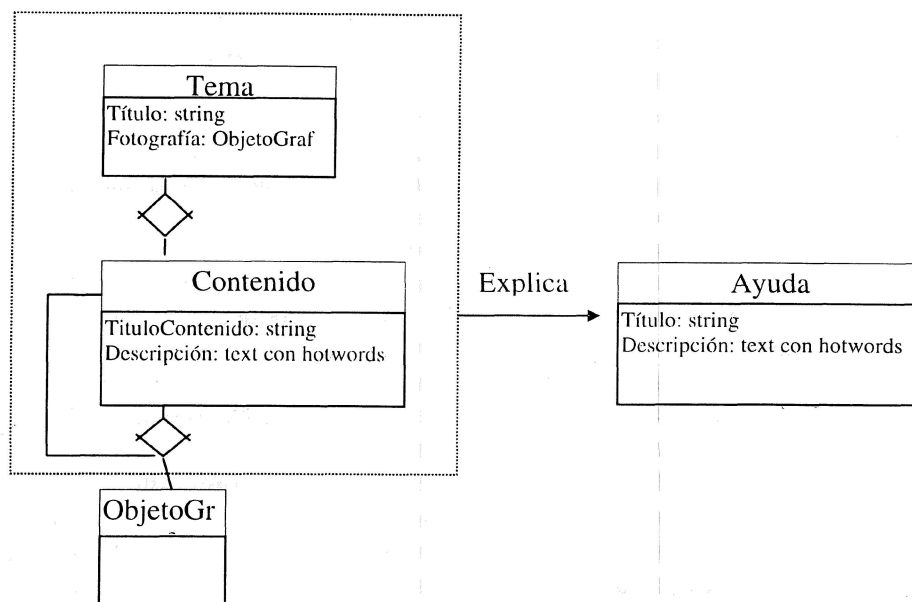


Fig. 5. Modelado del dominio en OOHDM

Todas estas especificaciones del diseño son relativamente compactas (si se hace excepción a los cartones de nodo, enlaces y contexto) y muy expresivas, excepto la especificación de la interfaz que, como podemos apreciar, se manifiesta de manera poco precisa o demasiado general. También debe notarse la continuidad existente entre las distintas etapas de diseño y resaltarse que entre el diseño conceptual y el navegacional el paso es prácticamente natural, mientras que el paso del modelo navegacional al diseño de interfaz es brusco, dado que se pasa a utilizar otro tipo de modelo y ahí se pierde un poco la continuidad del diseño.

Esta metodología tiene como desventaja que requiere de cierta sobrecarga para aprenderla, debido a los modelos que utiliza. Sin embargo, OOHDM contempla todos los aspectos que debimos modelar, es completa para nuestro caso, los modelos subyacentes son ricos en representación estructural y semántica, más su utilización sin una herramienta CASE se hace compleja a medida que crece el tamaño y la complejidad de la aplicación a modelar.

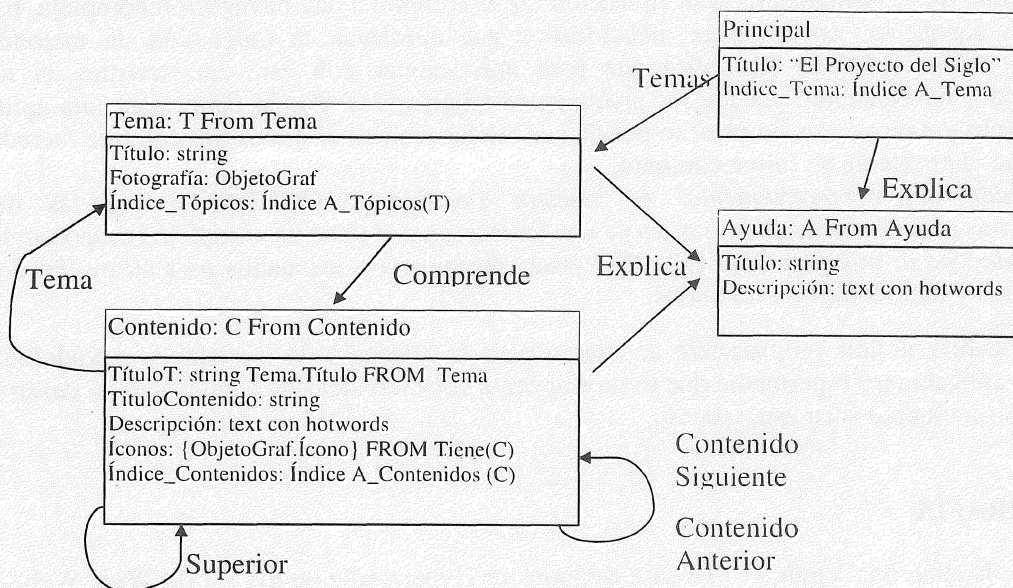


Fig. 6. Esquema navegacional en OOHDM

4. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se han presentados los resultados de una experiencia de diseño con 4 diferentes modelos y metodologías de desarrollo hipertexto. El objetivo principal que nos mueve era identificar fortalezas y debilidades de las mismas, para luego realizar un análisis comparativo que se enmarca en una propuesta de metodología evaluativa más amplia [Ortiz99].

En base a la experiencia y el análisis realizado, hemos identificado algunos criterios que deben ser tomados en cuenta por los autores para seleccionar un modelo o metodología de desarrollo de aplicaciones hipertexto para su utilización en un proyecto particular.

1. *Complejidad y otras características del dominio de la aplicación:* Para aplicaciones que tienen dominios complejos, vemos que OOHDM es la metodología que posee las más potentes primitivas para modelado del dominio. Al mismo tiempo permite al autor concentrarse en este problema al separarlo del resto de las actividades de diseño. Si la aplicación no tiene un dominio complejo hay que tomar en cuenta el alto costo de aprendizaje y la gran cantidad de documentación a generar con OOHDM. Si el dominio tiene estructuras repetitivas pueden ser considerados HDM o RMM como alternativas.
2. *Conocimientos previos del grupo de trabajo:* aprender a utilizar cada metodología conlleva cierto costo, sin embargo OOHDM posiblemente introduce una mayor complejidad, especialmente si los integrantes del grupo que lleva adelante el proyecto no tienen conocimientos del modelo de objetos. Incluso teniendo dichos conocimientos el paso a dar es grande, sobretodo en lo que a modelado de

interfaz se refiere. Si el grupo de trabajo tiene experiencia con la utilización de modelos del área de bases de datos (no orientados a objetos) puede resultar más fácil utilizar RMM o HDM.

3. *Complejidad de la navegación*: si la aplicación no se enfrenta a una navegación compleja, hay que tomar en cuenta los costos antes mencionados que conllevan la utilización de metodologías potentes. Se ha señalado [Rossi96] que para aplicaciones con esta característica el modelo conceptual y navegacional pueden ser prácticamente unificados. Por lo tanto, para una aplicación con dominio y navegación simples, se podría considerar realizar el desarrollo con una metodología que ataque el problema en forma conjunta.
4. *Complejidad del comportamiento*: en nuestra experiencia, OOHDM provee las mejores herramientas para controlar el desarrollo de una aplicación que tiene un complejo comportamiento.
5. *Complejidad de la interfaz*: Sólo OOHDM posee mecanismos adecuados para su diseño de entre los modelos y metodologías estudiados.

En el futuro cabría definir propiamente un marco para la selección de modelos y metodologías de desarrollo de aplicaciones hipermedia que tome en cuenta características de la aplicación a desarrollar y de los autores involucrados en este proceso.

5. BIBLIOGRAFÍA

- [Bieber97] Bieber, M.; Vitali, F. "Toward Support for Hypermedia on the World Wide Web". IEEE Computer, Vol. 30, N. 1, Enero 1997, también en <http://www.cs.unibo.it/~fabio/bio/papers/1997/IEEEC97/January/IEEEC0197.html>
- [Cern97] Cernuzzi, L.; Mareco, C. "Modeling Hypermedia Systems: a case study". Proceedings of Information Systems Analysis and Synthesis – SCI '97 / ISAS '97, Caracas, Julio 1997
- [Chris98] Christodoulou, S.; Styliaras, T.; Papatheodorou, T. "Evaluation of hypermedia application development and management systems". Proceedings of ACM Hypertext Conference 1998, pp 1-10
- [Garzo91] Garzotto, F., Paolini, P., y Schwabe, D. "HDM - A Model for the Design of Hypertext Applications". En Proceedings ACM Hypertext '91 (pp. 313-328).
- [Garzo93] Garzotto, F., Paolini, P., y Schwabe, D. "HDM - A Model-Based Approach to Hypertext Application Design". En ACM Transactions on Information Systems, Vol. 11, No. 1, Enero 1993, (pp. 1-26).
- [Garzo95] Garzotto, F., Mainetti, L., Paolini, P. "Hypermedia Design, Analysis, and Evaluation Issues". Communication of the ACM, Vol. 38 No. 8, (pp.74-86), Agosto 1995.
- [Halas94] Halasz, F., Schwartz, M. "The Dexter Hypertext Reference Model". En Communications of the ACM. Febrero 1994, Vol. 37, No. 2. (pp. 30-39).
- [Hardm93] Hardman, L., Bulterman, D.C.A. y van Rossum, G. "The Amsterdam hypermedia model: Extending hypertext to support real multimedia". En Hypermedia Vol. 5, No. 1, Julio 1993, (pp. 47-69).
- [Hardm94] Hardman, L., Bulterman, D.C.A. y van Rossum, G. "The Amsterdam hypermedia model: adding Time and Context to the Dexter Model". En Communications of the ACM Vol. 37, No. 2, Febrero 1994, (pp. 50-62).

- [Isako95] Isakowitz, T., Stohr, E. A., Balasubramanian, P. "RMM: A Methodology for Structured Hypermedia Design". En Communications of the ACM, Vol. 38, No. 8, agosto 1995, (pp. 34-44).
- [Lange94] Lange, D. "An Object-Oriented Design Method for Hypermedia Informations Systems". En Proceedings of the 27th Annual Hawaii International Conference on System Science, Enero 1994.
- [Ortiz99] Ortíz, S. "Evaluación de una Metodología de Desarrollo de Aplicaciones Hipermediales". Tesis de Ingeniería Informática, Universidad Católica "Nuestra Señora de la Asunción", Asunción - Paraguay, 1999.
- [Rossi96] Rossi, G. "An Objet-Oriented Method for Designing Hypermedia Applications". Tesis de Doctor em Informática. Departamento de Informática, Pontifícia Universidade do Rio de Janeiro, Brasil. En <http://www-lifia.info.unlp.edu.ar/~fer/oohdm/>
- [Schwa95] Schwabe, D., Rossi, G. "Building Hypermedia Applications as Navigational Views of Information Models". En Proceedings of the Hawaii International Conference on System Sciences, Hawaii, enero 1995.
- [Shnei92] Shneiderman, B. *Designing the User Interface*. Addison-Wesley Publishing Company, Reading, Massachusetts, segunda edición, 1992 .
- [Stott89] Stotts, D. P., y Furuta, R. "Petri-Net-Based Hypertext: Document Structure with Browsing Semantics". En ACM Transactions on Information Systems, Vol 7, No. 1, Enero 1989 (pp.3-29).
- [Tomp89] Tompa, F. "A Data Model for Flexible Hypertext Database Systems". En ACM Transaction on Information Systems, Vol. 7, No. 1, 1989 (pp 85-100).



