

# ***Metodologías de Concepción para Aplicaciones Hipermedia:***

## ***Análisis crítico***

Philippe Lopistéguy, Begoña Losada, Pantxika Dagorret Facultad de Informática,  
Universidad del País Vasco (jiplopop,jiblopeb)@si.ehu.es  
I.U.T. de Bayonne-Pays Basque Université des Pays de l'Adour Bayonne – France  
[Pantxika.Dagorret@iutbay.univ-pau.fr](mailto:Pantxika.Dagorret@iutbay.univ-pau.fr)

### ***Resumen***

*El principio de navegación se ha extendido a nivel del gran público gracias a su uso masivo en dominios tales como Internet y a la comercialización de software "amigable" para el desarrollo de aplicaciones con capacidades hipermedia. Recientemente, se han desarrollado modelos y métodos para el diseño de aplicaciones hipermedia con la finalidad de sugerir aproximaciones de diseño de acuerdo con las normas de calidad comúnmente aceptadas en ingeniería de software.*

*Este artículo realiza un estudio de estas aproximaciones, centrándose principalmente en la forma en que se efectúa la integración de las abstracciones hipermedia sobre el dominio de la aplicación. Primero, presentamos lo que, desde nuestro punto de vista, constituyen los puntos claves de la integración hipermedia, a continuación, efectuamos un análisis crítico de un conjunto significativo de métodos y finalmente, concluimos con una síntesis sobre sus contribuciones a la concepción de aplicaciones hipermedia.*

**Palabras-clave :** Hipermedia, Concepción, Metáfora

## **1. Introducción.**

El principio de navegación hipermedia entre elementos de información se ha extendido ampliamente a nivel del gran público gracias a los avances tecnológicos (puestos de información, PC multimedia, Internet,...), y se ha revelado como fácil de implementar gracias a herramientas software sencillas y potentes (HyperCard, AuthorWare, parsers HTML,...). Trabajos más recientes y específicos extienden este principio de navegación a dominios diferentes tales como las Bases de Datos [Garz93b], o la Integración de Aplicaciones [Gron94], [Filg95].

Paralelamente a estos trabajos, se han realizado esfuerzos en la especificación de modelos hipermedia para la definición de estructuras y funcionalidades comunes a la mayoría de los sistemas hipermedia. Existen diferentes modelos de representación [Garg88], [Afra90], [Lang90], [Hala94], siendo su punto común estar definidos a partir de funcionalidades abstractas basadas en una concepción similar de nodos y enlaces.

Más recientemente, el interés se ha volcado en los modelos y métodos de concepción de aplicaciones bien estructurados. El objetivo de estos trabajos es proponer aproximaciones incrementales de concepción de aplicaciones hipermedia según las reglas del arte: concepción bien documentada y reusable, aplicación sencilla y fácil de

mantener... Se puede considerar que la situación actual de la concepción de aplicaciones hipermedia es similar a la denominada crisis del software en la que los programas y materiales permitían la realización rápida de aplicaciones efectivas mientras que su concepción pecaba de falta de metodología. El resultado eran aplicaciones poco estructuradas y difíciles de mantener. Las técnicas de la programación estructurada y modular -y sus conceptos subyacentes- han proporcionado una solución a esta situación.

Este artículo se inscribe dentro de esta línea de preocupación y se interesa particularmente en el proceso que se sigue en la integración de abstracciones hipermedia en la concepción de una aplicación.

La primera sección del artículo expone nuestro punto de vista sobre aspectos fundamentales de la concepción de aplicaciones hipermedia, la sección siguiente presenta una descripción crítica de modelos y métodos significativos de concepción hipermedia, y la última parte efectúa una síntesis sobre las aportaciones de estos métodos a la concepción de aplicaciones hipermedia.

## **2. Puntos clave del proceso de hiperización.**

Hemos constatado [Lopi96] que en la concepción de una aplicación, la adquisición de funcionalidad hipermedia consiste en dotar a determinados elementos del dominio de la aplicación de "metáfora hipermedia"<sup>1</sup>. Consideramos este aspecto como el punto clave de la concepción de aplicaciones hipermedia y en esta sección abordamos las etapas que caracterizan su realización.

### **2.1 Dotar a los elementos de la aplicación de metáfora hipermedia.**

Consideramos que toda concepción de aplicación hipermedia supone, entre otras, las tres tareas siguientes cuya combinación consigue la integración del paradigma hipermedia.

- 1 - Identificar/definir elementos del dominio de la aplicación.
- 2 - Identificar/definir elementos hipermedia.
- 3 - Identificar/definir la relación que les une.

Entendemos por elemento del dominio de la aplicación todo componente (objeto, entidad, relación, atributo, evento, operación, abstracción, ...) especificado en el curso del ciclo de vida de la aplicación, y por elemento hipermedia todo componente (nodo, enlace, ancla, método, evento, ...) propio de los modelos hipermedia.

Aunque estas tres tareas estén numeradas, su realización no implica el respeto de este orden y la realización de una no implica el comienzo de la siguiente. Sin embargo, desde un punto de vista metodológico, el estudio preliminar de la organización del dominio es necesario ante cualquiera de estas tareas.

### **2.2 Los elementos del dominio de la aplicación.**

---

<sup>1</sup> Los elementos del dominio de la aplicación son dotados de funcionalidades hipermedia que les proporciona un comportamiento hipermedia.

En cuanto a la organización del dominio, se han propuesto varios modelos, en función del método empleado; modelo objeto para OMT [Rumb91], modelo Entidad-Relación y diagramas de flujo de datos o modelos conceptuales de tratamiento para métodos estructurados como Merise o Métrica. Globalmente, los elementos especificados en estos modelos son clases, entidades, relaciones, procesos, mensajes o eventos y según nuestra aproximación, estos elementos son, sin restricción, candidatos potenciales a ser dotados de metáfora hipermedia, lo que extiende el propósito general de las aproximaciones más conocidas como veremos a continuación.

Los elementos de base del dominio son el origen de la integración del paradigma hipermedia, teniendo en cuenta que no todos tienen sistemáticamente "necesidad" de una imagen hipermedia. Cuando la identificación del elemento no es adecuada, puede ser necesario definir nuevos elementos de aplicación por mecanismos de especialización o de combinación-agregación. Esta actividad constituye la primera de las tareas que hemos identificado como punto clave del proceso de hiperización.

El modelo resultante de esta actividad se denomina modelo enriquecido del dominio.

## 2.3 Los elementos hipermedia.

Generalmente, los elementos hipermedia más comunes son aquellos reagrupados bajo el término de hiperobjeto, que se atribuye a los nodos, enlaces y anclas así como a sus múltiples variantes tales como enlaces mono-bi-multi-direccionales o nodos simples o compuestos. Otras estructuras más complejas pero igualmente clásicas son menos utilizadas, como índices, visitas guiadas (*guided tour*), históricos, caminos (*trail, path*), mapas (*map*), navegadores (*browser*), marcadores (*bookmark, landmark*) o incluso anotaciones. Sin embargo, aunque la semántica de estos hiperobjetos esté bien establecida, el detalle de sus características difiere a menudo en cada aplicación.

Además de los hiperobjetos, otros elementos hipermedia como métodos (CrearNodo asociado al elemento de aplicación NuevaEntidad), atributos (ContenidoNodo asociado al elemento de aplicación NombreApellido) o eventos (CerrarNodo asociado al elemento de aplicación TimeOut) pueden ser, a nuestro modo de ver, interesantes para el diseñador.

Por lo tanto, es necesario que éste pueda definir el elemento hipermedia que corresponda mejor a la metáfora hipermedia que desea dar al elemento de aplicación. Puede hacerlo bien precisando las características de un hiperobjeto (accesibilidad, comportamiento, ...), bien definiendo claramente un elemento hipermedia (evento, atributo, composición de elementos hipermedia, ...). El modelo hipermedia enriquecido así obtenido podrá resultar de un modelo inicial adaptable y extensible. El modelo resultante de esta actividad se denomina modelo hipermedia enriquecido. Remarquemos que el enriquecimiento del modelo hipermedia podrá resultar del mecanismo de correspondencia que estamos describiendo.

En efecto, si el dominio de la aplicación es el dominio hipermedia mismo, gracias a este mecanismo de correspondencia será posible, por ejemplo, asociar una lista de nodos a un nodo, obteniendo así nodos cuyo contenido son asimismo nodos. Esta técnica autoriza la definición de elementos hipermedia a partir de otros elementos hipermedia y en el caso de nodos compuestos o de índices, esto evita la creación de

enlaces ambiguos entre el nodo compuesto y los nodos componentes, conforme a las recomendaciones de [Hala91].

## **2.4 La metáfora hipermedia.**

El principio de correspondencia entre elementos de aplicación y elementos hipermedia es subyacente a los métodos presentados en la sección tercera, incluso si no se expresan en términos tan genéricos como en nuestro análisis. Nos hemos esforzado a abstraer mecanismos tipo y, además del principio de metáfora, hemos identificado aspectos de Correspondencia y de Proceso de concepción que detallamos en esta sección.

### **2.4.1 Definición de correspondencias.**

Generalmente, los tipos de correspondencia propuestos al diseñador son específicos de cada método, es decir que es difícil encontrar correspondencias similares en dos métodos diferentes.

La correspondencia se define para tipos particulares de elementos del modelo del dominio, como por ejemplo Relación <-> Enlace o bien para elementos del dominio enriquecido construidos según mecanismos de fragmentación o de composición propios del método, como por ejemplo Trozo <-> Nodo [Isak95]. Así, en el curso de una de las etapas del método, el diseñador identifica/define elementos del dominio, y les atribuye una de las metáforas tipo entre las correspondencias propuestas.

Cuando los métodos tienen una herramienta de concepción asociada, los tipos de correspondencia específicas y preestablecidas toman todo su interés puesto que el trabajo de implementación está parcial o totalmente efectuado por la herramienta. Sin embargo, la potencialidad hipermedia está limitada y no se explota en su totalidad.

Aunque el principio de los tipos de correspondencia preestablecidos sea interesante para el diseñador, la posibilidad de seleccionar un elemento de aplicación en lugar de otro es un punto crítico. Diferentes trabajos tienden a considerar la concepción hipermedia como la "ciencia de las relaciones", en los que el punto clave reside en las relaciones que unen los elementos del dominio. Así, según esta aproximación, la metáfora de enlace está asociada al concepto de relación [Bieb96].

En cuanto a la tarea de selección de elementos del dominio, puede ser interesante entonces poder identificar diferentes tipos de relaciones. Por ejemplo:

- relaciones de proceso, que dan acceso directo a operaciones y procesos.
- relaciones estructurales, que dan acceso a objetos de la estructura interna de una aplicación.
- relaciones de metainformación, que dan acceso a los parámetros y a la información descriptiva de los objetos de una aplicación.
- relaciones de ocurrencia, que dan acceso a todas las apariciones de un elemento dado.

Para cada una de estas relaciones hay que proponer una correspondencia -o metáfora- de enlace específica.

En el caso de aplicaciones con capacidad de *authoring* hipermedia, el usuario es a su vez diseñador. Incluso si la aplicación no lo especifica explícitamente, permite al usuario, bajo forma de menú u otra, seleccionar un elemento del dominio y relacionarlo con un elemento hipermedia (ancla, enlace, nodo, anotación<sup>2</sup>). Podemos citar como ejemplo MetaEdit+ que es una herramienta CASE [Oina96] dotada de capacidades hipermedia para documentar, relacionar y confrontar diferentes decisiones de concepción.

De esta forma, el mecanismo de metáfora permanece subyacente, las tareas para llevarlo a cabo son implementadas bajo forma de servicios y las correspondencias se realizan en el momento en que surge la necesidad. Es decir que la selección del elemento del dominio no es inducido directamente por el modelo conceptual del dominio.

## **2.4.2 Proceso de establecimiento de la correspondencia.**

Hemos podido observar que, según los modelos y los métodos, el impacto de la correspondencia puede afectar a la globalidad de los elementos del dominio o solamente a una parte de ellos.

Cuando la correspondencia se aplica al conjunto de los elementos, el método puede ser calificado de impositivo y enfocado a la obtención de aplicaciones tipo. En este caso, el proceso de concepción propone etapas con la finalidad de concebir aplicaciones hipermedia estereotipadas. La correspondencia es sistemática, el modelo del dominio está enriquecido con elementos predefinidos y cuya correspondencia está preestablecida. La ventaja de este proceso es ofrecer una línea de trabajo precisa que facilita la concepción e impide iniciativas "peligrosas".

Por otra parte, cuando la correspondencia se aplica a una parte de los elementos, el método puede ser calificado de abierto y enfocado a la obtención de aplicaciones dotadas de funciones variadas. Un ejemplo de aplicaciones de este tipo son los sistemas tutores inteligentes que dan acceso libre a la información. En efecto, sus capacidades hipermedia y de tutoría deben cohabitar desde el nivel conceptual.

La finalidad que persigue es permitir la integración de funcionalidades hipermedia durante el proceso de concepción. En este caso, la correspondencia entre elementos del dominio<sup>3</sup> y elementos hipermedia no es sistemática, admite variaciones. La libertad de elección es preferible, aunque también deben poder ser propuestos modos de correspondencia tipo a fin de facilitar la tarea y de evitar errores conceptuales.

Así, hacemos una distinción entre las aplicaciones concebidas totalmente según una aproximación hipermedia, y las aplicaciones que disponen de funcionalidades hipermedia en las que el paradigma hipermedia no es tenido en cuenta más que en una parte de su proceso de concepción.

---

<sup>2</sup> Hacemos notar que, aún no siendo un elemento hipermedia típico, la anotación es un elemento difícil de integrar en una etapa de análisis conceptual hipermedia.

<sup>3</sup> Por dominio entendemos el de la aplicación en su globalidad y no el "dominio" a enseñar, en el sentido de los tutores inteligentes.

### **3. Diferentes aproximaciones de concepción de aplicaciones hipermedia.**

El proceso de concepción de una aplicación puede ser implícita o bien explícita y en este último caso, los diseñadores siguen o bien un proceso propio (*ad-hoc*), o bien una aproximación genérica. En nuestro estudio, hemos retenido los procesos de este último tipo. En esta sección, repasamos brevemente, para los lectores no familiarizados con el dominio, la aproximación y la terminología de los modelos y métodos que nos han parecido significativos. Posteriormente, efectuamos un análisis crítico de cada uno basado en los puntos comentados en la sección precedente; esto es, en términos de dominio enriquecido, hipermedia enriquecido, correspondencia preestablecida y proceso abierto o impositivo. El lector experto se centrará en las partes de análisis.

#### **3.1 HDM : A Model-based Approach to Hypertext Application Design.**

HDM [Garz93a] constituye un primer paso en la definición de un método descendente de concepción de aplicaciones hipertexto. Ha sido la fuente de inspiración de los métodos RMM [Bala94], [Isak95] y OOHDM [Schw95]. El modelo HDM no se interesa en la concepción del contenido de los nodos (*authoring in the small*), se centra únicamente en la concepción topológica de las aplicaciones (*authoring in the large*).

##### **3.1.1 Los elementos del modelo.**

HDM propone un conjunto de elementos que permiten al diseñador especificar una aplicación. Estos elementos son las entidades, los componentes, las perspectivas, las unidades y los enlaces. El término entidad se ha extraído del modelo Entidad-Relación, pero se ha extendido para poder representar una estructura compleja que contenga enlaces y una semántica de navegación internas.

Una entidad es una jerarquía de componentes que "heredan" las propiedades de la entidad y que no pueden existir más que como partes de la entidad. Las perspectivas permiten representar la multiplicidad de las presentaciones de un mismo contenido de información (presentación en diferentes lenguas de un mismo texto,...). Las unidades son los depósitos de la información contenida en la aplicación. Una unidad representa un fragmento del contenido de una entidad presentada bajo una perspectiva particular. Un componente es una abstracción para diseñar un conjunto de unidades que representan todas las perspectivas de un mismo contenido de información. Si una entidad tiene dos perspectivas, todos sus componentes tendrán dos perspectivas.

En el modelo, los enlaces hipermedia tienen un papel doble; el de representar la semántica del dominio y el de permitir la navegación. Para esto, HDM define tres tipos de enlaces: enlaces estructurales entre componentes de una misma entidad, enlaces de perspectiva entre unidades de un mismo componente, y enlaces de aplicación entre entidades o componentes. Las ventajas aportadas por esta clasificación son una utilización más consistente de los enlaces, esquemas de navegación definidos de antemano, una semántica de navegación por defecto y la puesta en marcha, desde la fase de desarrollo, de mecanismos como la derivación de enlaces. Durante la concepción de

la aplicación, los enlaces de perspectiva y estructurales se derivan automáticamente del esquema (de la estructura de las entidades).

La aportación principal de HDM2 [Garz93c] con respecto a HDM consiste en la añadidura al modelo de estructuras de accesos como los índices y las visitas guiadas.

### **3.1.2 Análisis crítico.**

Por su condición de modelo, HDM no se centra en la especificación del dominio de la aplicación. Su aportación se sitúa más allá, proponiendo un modelo que sirva de base a la aplicación de correspondencias tipo. En efecto, los elementos (entidad, componente, perspectiva, unidad, enlace) constituyen un lenguaje de especificación que permite al diseñador definir un modelo enriquecido del dominio. Mediante las correspondencias preestablecidas por HDM, este modelo enriquecido define la estructura navegacional de la aplicación.

Entre los elementos del dominio, distinguimos las entidades y componentes (que permiten definir la estructura de la información a hiperizar) de las perspectivas y unidades (que permiten definir las variaciones de semántica de la información). El sistema de correspondencias preestablecidas asocia estos elementos al elemento hipermedia Nodo. Las relaciones recogidas de las primitivas propias de HDM (entidad, componente, unidad) son estructurales, y se les asocia la metáfora preestablecida de Enlace. Sin embargo, el concepto de enlace se deja disponible para ser asociado igualmente a relaciones de semántica o de aplicación.

Como conclusión, podemos decir que HDM propone un modelo enriquecido para especificar el dominio de la aplicación. A partir de este dominio enriquecido, y gracias a un mecanismo de correspondencia preestablecido, se describe la estructura navegacional de la aplicación. Por otra parte, si bien es cierto que HDM2 introduce estructuras de acceso, el modelo hipermedia subyacente es mínimo y no enriquecido. Los calificativos de abierto o de impositivo - que se aplican al proceso de concepción - no son pertinentes para HDM ya que éste es sólo un modelo. Las dos subsecciones siguientes se consagran a dos métodos inspirados de este modelo; uno impositivo, el otro abierto.

## **3.2 RMM : Relationship Management Methodology.**

El método está definido como un proceso de análisis, diseño y desarrollo de aplicaciones hipermedia cuya estructura es estable y cuyo contenido sufre modificaciones frecuentes. Los elementos principales de este método son el modelo ER (*Entity-Relationship*) y el modelo RMDM (*Relationship Management Data Model*) basado en el modelo HDM [Garz93a,c]. Las aportaciones con respecto a HDM residen en la descripción detallada de las etapas diferentes del proceso de concepción por una parte, y por otra parte, en la manera en que los diferentes elementos del modelo deben ser utilizados.

### **3.2.1 El modelo RMDM.**

El modelo propone un lenguaje que permite describir los objetos del dominio, sus interrelaciones y los mecanismos de navegación hipermedia de la aplicación.

Los objetos del dominio se definen con la ayuda de entidades, atributos y relaciones asociativas. El modelo introduce el concepto de trozo (*slice*) con el fin de modelizar los aspectos unidos a la presentación de las entidades. Un trozo corresponde a un subconjunto de atributos de una misma entidad destinados a ser presentados de forma agrupada. La navegación se modeliza con la ayuda de primitivas de acceso; enlaces estructurales (unidireccional y bidireccional) que permiten especificar la navegación entre trozos, y visita guiada condicional, índice condicional y agrupación, que permiten especificar la navegación entre entidades.

El esquema completo del dominio y de la navegación de la aplicación se denomina esquema RMDM y se obtiene como resultado de las tres primeras etapas del método.

### **3.2.2 Las etapas del método.**

La primera etapa consiste en representar los objetos del dominio con la ayuda del modelo Entidad-Relación ampliado con relaciones asociativas (aquellas que permiten representar caminos navegacionales entre entidades puestos en evidencia en la fase de análisis).

La segunda etapa consiste en determinar la presentación del contenido de las entidades de la aplicación así como su modo de acceso. El esquema obtenido como resultado de esta etapa se denomina esquema E.R+. Se trata de un esquema Entidad-Relación en el que cada entidad ha sido reemplazada por su esquema de entidad. Un esquema de entidad está constituido por nodos (los trozos) unidos por relaciones estructurales.

La tercera etapa consiste en definir los caminos de navegación inducidos por las relaciones asociativas del esquema E-R+. A continuación, es posible definir estructuras de acceso de alto nivel (agrupaciones), lo que permite dotar a la aplicación de accesos jerárquicos a niveles diferentes de los trozos de información. El esquema RMDM resultante se obtiene añadiendo al esquema E-R+ las agrupaciones y caminos navegacionales definidos en esta etapa.

Las cuatro etapas restantes consisten en -1- definición del protocolo de conversión de elementos del diagrama RMDM en objetos de la plataforma de desarrollo, -2- concepción del interfaz usuario, -3- concepción del comportamiento en ejecución y -4- construcción del sistema y test.

### **3.2.3 Análisis crítico.**

De acuerdo con HDM, RMM define correspondencias tipo. Propone el modelo Entidad-Relación para especificar el dominio y lo enriquece introduciendo los conceptos de trozo y de relación estructural para los cuales una metáfora hipermedia es preestablecida. Los trozos y las entidades se asocian con nodos y las relaciones asociativas (entre entidades) y las relaciones estructurales (entre trozos) se asocian con enlaces.

El modelo del dominio se enriquece, por lo tanto, con dos tipos de elementos preestablecidos que tienen una correspondencia clara en términos hipermedia.

En RMM, el modelo hipermedia retoma los elementos enlace, índice y visitas guiadas de HDM enriqueciéndolos con capacidades condicionales. Sin embargo, el método no permite al diseñador definir elementos hipermedia propios que tengan capacidades específicas ya que impone la utilización de metáforas preestablecidas.

Calificamos, por tanto, a RMM de método impositivo pues da precisiones sobre todas las etapas a realizar, desde la concepción del esquema del dominio hasta el desarrollo de la aplicación hipermedia.

### **3.3 OOHDM : Object Oriented Hypermedia Design Methodology.**

Al igual que RMM, este método se inspira en el modelo HDM. Lo que lo distingue claramente de RMM es el proceso de concepción orientado a objetos. OOHDM [Schw96] tiene cuatro etapas, cada una centrada en un aspecto de la concepción.

#### **3.3.1 Los elementos del dominio y etapas del método.**

Cada etapa de la concepción define un esquema objeto específico en el que se introducen nuevos elementos (clases). En la primera etapa, el análisis del dominio consiste en establecer un esquema conceptual en términos de clases, relaciones y subsistemas. Para ello, el método propone el modelo objeto de OMT. En la segunda etapa, el diseñador define clases navegacionales tales como nodos, enlaces, índices y visitas guiadas inducidas del esquema conceptual. Los enlaces derivan de relaciones y los nodos representan ventanas lógicas (*views*) sobre las clases conceptuales. A continuación, el diseñador describe la estructura navegacional en términos de contextos navegacionales. Estos contextos definen agrupaciones -en el sentido de HDM- de objetos navegacionales (nodos, enlace,...). Durante esta etapa, es posible adaptar los objetos navegacionales para cada contexto, de forma similar a las perspectivas de HDM. La tercera etapa, dedicada a la especificación del interfaz abstracto, describe los objetos de interfaz y los asocia con objetos de navegación. Por fin, la última etapa, consagrada a la implementación, hace corresponder los objetos de interfaz con objetos de implementación.

#### **3.3.2 Análisis crítico.**

OOHDM no propone un modelo enriquecido para el dominio de la aplicación, lo que deja al diseñador libre para elegir el modelo de especificación del dominio. Por contra, el modelo hipermedia está definido en dos niveles de abstracción; las clases navegacionales y los contextos navegacionales.

En el momento de la especificación de las clases navegacionales es cuando el diseñador define las correspondencias. Aunque OOHDM sugiere algunas, no impone metáforas preestablecidas tan sistemáticamente como RMM. Los nodos inducidos de las clases del modelo del dominio y los enlaces inducidos de las relaciones del modelo del dominio pueden ser precisados. El segundo nivel está consagrado a la especificación de

la navegación, expresada exclusivamente sobre los objetos navegacionales (no sobre los elementos del modelo del dominio), lo que constituye un mecanismo que permite enriquecer el modelo hipermedia.

Aunque los ejemplos que ilustran el método sean siempre del mismo tipo, calificamos OOHDM de método abierto porque, por una parte, el modelo del dominio no viene impuesto y por otra parte, el soporte en objetos del método permite la especialización de las clases navegacionales y de los contextos navegacionales. En efecto, el objetivo de OOHDM es cubrir la concepción de todo tipo de aplicaciones hipermedia.

### **3.4 EORM : Enhanced Object-Relationship Model.**

El método se define como un proceso informal de concepción de sistemas de información dotados de funcionalidades hipermedia. El proceso propuesto consiste en llegar a esquemas EORM que sirvan de especificación para la concepción de aplicaciones.

#### **3.4.1 El enlace, elemento clave del modelo.**

Los esquemas EORM se constituyen a partir de un modelo conceptual completado con clases de enlaces que especifican sus aspectos estáticos y dinámicos. Se proponen clases tipo aunque el diseñador define preferiblemente clases enlace *ad-hoc*, según sus necesidades. Este aspecto constituye la filosofía y característica principal de EORM.

#### **3.4.2 Las etapas del método.**

La primera etapa consiste en un análisis del sistema con la ayuda de un método orientado a objetos (OMT), que define la estructura de las informaciones (modelo objeto), su comportamiento (modelo dinámico) y sus interrelaciones.

La segunda etapa consiste en crear el esquema EORM de la aplicación a partir de los elementos proporcionados por la etapa anterior. Este esquema especifica las relaciones de interacción de la aplicación, en donde las relaciones entre objetos son vistas como caminos de interacción semánticamente ricos (es decir, que tienen una estructura y un comportamiento), más que como simples relaciones estructurales.

De esta forma, las relaciones definidas en el modelo objeto de OMT pueden ser representadas por clases de enlaces hipermedia. Estas clases añaden a las relaciones del modelo objeto la semántica navegacional de la aplicación. Están organizadas en una jerarquía de herencia propuesta por el método bajo la forma de una biblioteca de clases. La semántica relativa a las propiedades hipermedia de las relaciones encuentra, por tanto, una representación en EORM bajo la forma de clases.

Finalmente, la tercera etapa consiste en transformar los esquemas EORM en código, en salvaguardarlos en una Base de Datos Orientada a Objetos, y en elaborar formularios de consulta de las clases con la ayuda de un editor gráfico de interfaces.

#### **3.4.3 Análisis crítico.**

El modelo EORM no enriquece el dominio con nuevos conceptos. Los elementos candidatos a la metáfora hipermedia se reducen por tanto, a las clases y relaciones del modelo OMT.

Por otra parte, toda la semántica de las relaciones de la aplicación se expresa por medio de enlaces hipermedia reagrupados en una jerarquía de clases. Es interesante resaltar que el comportamiento definido sobre los enlaces permite tener en cuenta una parte de la semántica de la navegación de la aplicación. Asimismo, permite alterar la propia estructura navegacional de la aplicación mediante operaciones de creación, destrucción o de modificación de elementos hipermedia. Está claro que la aproximación adoptada por EORM ha consistido en concentrarse en el enriquecimiento de los elementos hipermedia.

En cuanto al establecimiento de correspondencias entre elementos del dominio y elementos hipermedia, el método EORM da una cierta libertad en la elección de las metáforas para los enlaces. En efecto, precisa que no toda relación del dominio está determinada en convertirse en enlace navegacional y que los enlaces se deben adaptar a las necesidades del análisis.

Por contra, el proceso de realización de la correspondencia es impositiva, en el sentido de que una entidad siempre se transforma en nodo y una relación en enlace. Esto está justificado por el método puesto que considera que las relaciones son las claves de la hiperización.

### **3.5 El modelo Trellis.**

Trellis [Furu90] es un modelo generalizado de hipermedia. No propone un método de concepción. Indica que un sistema hipermedia está compuesto por tres elementos distintos; el contenido de la información, la estructura navegacional y el control navegacional (esto es, el control de la dinámica de la aplicación).

#### **3.5.1 Los elementos del modelo.**

Este modelo propone las redes de Petri [Reis85], esquematizadas por un grafo orientado construido a partir de *places*, de *transitions* y de *tokens*, para especificar las características de interacción de una aplicación. Los elementos *places* representan los contenidos de información, los *tokens* indican cuáles son las informaciones en curso de presentación, los elementos *transitions* representan el encadenamiento de las presentaciones y el grafo en su conjunto describe la estructura navegacional de la aplicación.

Cuando los elementos *places* situados a continuación de una *transición* se convierten en *token*, se dice que la *transition* es activable. El usuario puede seleccionarla para acceder (navegar) a los contenidos de información colocados más abajo. Las *transitions* están dotadas de atributos (tiempo de espera, tiempo de liberación) que permiten al diseñador asociarles características temporales. El tiempo de espera es el lapso de tiempo que indica al cabo de cuánto tiempo una *transition* activable puede ser seleccionada por el usuario. El tiempo de liberación es el lapso de

tiempo que indica al cabo de cuánto tiempo una *transition* seleccionable será franqueada automáticamente por el sistema en caso de no acción por parte del usuario. El modelo permite por lo tanto, una especificación cómoda de la sincronización, concurrencia y secuenciación de presentación de contenidos de información.

El modelo se define como genérico puesto que extiende la noción de presentación de la información a la de ejecución de programa. Así, el control de la navegación especifica la sincronización, la concurrencia y la secuenciación de programas en sentido amplio.

### **3.5.2 Análisis crítico.**

Trellis no propone un modelo enriquecido para especificar el dominio. Como se centra en las especificaciones navegacionales, propone un modelo gráfico que ilustra su aproximación.

Define correspondencias preestablecidas. El elemento *place* (nodo) modeliza a la vez el contenido y el tipo de la presentación. Las *transitions* (enlaces) corresponden a enlaces multidireccionales condicionales cuya activación no es posible más que si todos los nodos origen se presentan y cuya selección tiene por efecto presentar todos los nodos destino.

El modelo hipermedia está fijado pero constituye según el autor un mecanismo elemental que permite especificar elementos complejos.

Aunque el modelo no propone un proceso de concepción, está claro que se distingue de los modelos precedentes para los que la hiperización se deriva esencialmente del modelo del dominio. Con Trellis, el diseñador deberá prestar una atención particular al control de la navegación y a la sincronización ya que esto constituye, según el modelo, los elementos clave de la hiperización.

## **4. Conclusión y futuros trabajos.**

Desde hace algún tiempo, van apareciendo métodos de concepción de aplicaciones hipermedia. Tienen por objetivo poner a la disposición del diseñador un proceso -a veces incluso una herramienta/taller- fundado en un modelo, que le permite construir aplicaciones hipermedia que tienen en cuenta criterios de calidad adoptados comúnmente en programación.

Hemos analizado un conjunto de métodos y modelos subyacentes (HDM, RMM, OOHD, EORM, modelo de Trellis) que, por sus puntos comunes y divergentes, representan, según nuestro modo de ver, una muestra significativa del estado del arte en este dominio. El resultado que se repite en este estudio es que estos métodos no definen de forma satisfactoria las etapas que constituyen un proceso completo y genérico de cualquier tipo de aplicación hipermedia.

En efecto, es difícil hacer una distinción entre los conceptos relativos al modelo del dominio de la aplicación y aquellos del modelo hipermedia. Así, el dominio de la aplicación se especifica a menudo con la ayuda de términos de semántica híbrida, de tal forma que las responsabilidades hipermedia son difusas y difícilmente identificables. Como consecuencia de esta ambigüedad, constatamos que los mecanismos de

correspondencia entre los conceptos del dominio de la aplicación y los del dominio hipermedia son propios de cada método.

Para realizar este estudio, hemos analizado los métodos y modelos según tres puntos que consideramos claves en la concepción de aplicaciones hipermedia, esto es: - 1- definición de los elementos del dominio de la aplicación, -2- definición de los elementos hipermedia y, -3- definición de la metáfora hipermedia.

Según estos puntos clave, el método RMM propone enriquecer el modelo del dominio con conceptos como trozo, componente, unidad, para atribuirles a continuación metáforas hipermedia. Por contra, los métodos OOHDm y EORM suponen un dominio modelizado, con OMT, y proponen mecanismos para atribuir metáforas hipermedia a los elementos del modelo conceptual. El modelo Trellis y HDM no se interesan por este aspecto ya que son modelos hipermedia.

En cuanto a los elementos hipermedia, HDM y RMM proponen elementos preestablecidos y fijos. El modelo EORM propone exclusivamente un mecanismo de especialización de enlaces, mientras que Trellis propone elementos de base aptos para la construcción de elementos más complejos.

Igualmente, hemos constatado que el criterio de hiperización varía según los métodos; para el modelo Trellis se trata de la estructura navegacional y de los mecanismos de sincronización, para EORM se trata de las relaciones, y finalmente, para RMM y OOHDm se trata de estructuras de datos - mientras que HDM (su modelo subyacente) no precisa nada en este sentido-.

Se deduce de este estudio que cada uno de los métodos se ha interesado más particularmente en uno u otro de los puntos clave, lo que nos incita a considerar que la importancia de estos puntos es significativa en el proceso de concepción de aplicaciones hipermedia y que pueden servir, por lo tanto, como referencia para futuros trabajos.

Actualmente, nuestra actividad se centra en la especificación de un taller hipermedia que soporte la integración libre de funcionalidades hipermedia. El objetivo es proponer al desarrollador herramientas de prototipado poderosas que le permitan explorar posibilidades nuevas para mezclar el dominio hipermedia con otros dominios. Este taller se construirá sobre HyperFrame [Rodr97], un *hypermedia software framework* fundamentado en los tres puntos clave del proceso de integración. HyperFrame define los componentes hipermedia más elementales, sus interdependencias, mecanismos de construcción hipermedia así como un mecanismo de correspondencia (*mapping*), que permite asociar todo elementos hipermedia con un elemento del dominio externo. Este tipo de herramienta constituye, a nuestro entender, un soporte informático bien adaptado a la puesta en obra de la hiperización según los tres puntos claves que hemos definido.

## 5. Bibliografía.

- [Afra90] *A hypertext model supporting query mechanisms*, F. Afrati, C.D. Koutras, Proceedings of the European Conference on Hypertext, Editions Cambridge University Press, The Cambridge series on Electronic Publishing, 1990.

- [Bala94] *Designing Hypermedia Applications*, P. Balasubramanian, Tomás Isakowitz, Edward A. Stohr, Proceedings International Conference on System Sciences (HICSS), IEEE Computer Society Press, 1994.
- [Bieb96] *What Every Information Systems Developer Should Know About Hypertext*, M. Bieber, Proceedings Hypertext'96 - Workshop on Incorporating Hypertext Functionality into Software Systems, Washington, USA, 1996.
- [Filg95] *Towards an Open Hypermedia Systems*, J.M. Filgueira, I. Usandizaga, T. Pérez, Proceedings INFORSID'95, Grenoble, France, 1995.
- [Furu90] *Generalizing Hypertext : Domains of the Trellis Model*, R. Furuta, P. David Stotts, Technique et Science Informatique (TSI), Edition. Afcet-Bordas, vol. 9, no. 6.
- [Garg88] *Abstraction mechanisms in hypertext*, P.K. Garg, Communications of the ACM, vol. 31, no. 7, 1988.
- [Garz93a] *HDM. A model-based approach to hypertext application design*, F. Garzotto, P. Paolini, D. Schwabe, ACM Transactions on Information Systems, vol. 11, no. 1, 1993.
- [Garz93b] *Navigation patterns in hypermedia databases*, Proceedings Hawaii International Conference on System Sciences (HICSS), IEEE Computer Society Press, 1993.
- [Garz93c] *HDM2 : Extending the E-R Approach to Hypermedia Application Design*, F. Garzotto, L. Mainetti, P. Paolini, Proceedings International Conference on the Entity-RelationShip Approach, Arlington, USA, 1993.
- [Gron94] *Design Issues for a Dexter-based hypermedia system*, K.Gronbaek, R.H. Trigg, Communications of the ACM, vol. 37, no. 2, pp. 40-49, 1994.
- [Hala91] *Seven Issues : Revisited*, F. Halasz, Report SSL-91-149, Systems Science Laboratory, Palo Alto Research Center, Palo Alto, CA, 1991.
- [Hala94] *The Dexter hypertext reference model*, F. Halasz, M. Schwartz, in Communications of the ACM, vol. 37, no. 2, 1994.
- [Isak95] *RMM : a Methodology for Strutured Hypermedia Design*, Tomás Isakowitz, Edward A. Stohr, P. Balasubramanian, in Communications of the ACM, vol. 38, no. 8.
- [Lang90] *A formal model of Hypertext*, D.B. Lange, Proceedings of the Hypertext Standardization Workshop, National Institute of Standards and Technology Editions, 1990.
- [Lopi96] *Experiences and Reflection on the Use of a Hypermedia Framework for Hypermedia Functionality Integration*, P. Lopistéguy, I. Usandizaga, J.M. Filgueira, Proceedings Hypertext'96 - Workshop on Incorporating Hypertext Functionality into Software Systems, Washington, USA, 1996.

- [Oina96] *Hypermedia functionality in modeling tools*, H. Oinas-Kukkonen, Proceedings Hypertext'96 - Workshop on Incorporating Hypertext Functionality into Software Systems, Washington, USA, 1996.
- [Reis85] *Petri Nets : An Introduction* , W. Reisig. Springer-Verlag, 1985.
- [Rodr97] *Proceso y arquitectura para la construcción de aplicaciones hipermedia*, D. Rodríguez, J.M. Filgueira, P. Lopistéguy, Proceedings Jornadas sobre Ingeniería del Software, San Sebastián, España, 1997.
- [Rumb91] *Object Modeling Modeling and Design*, J. Rumbauch, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen. Englewood Cliffs, NJ: Prentice Hall, 1991.
- [Schw95] *The Object Oriented Hypermedia Design Model*, D. Schwabe, G. Rossi, Communications of the ACM, vol. 38, no.8, 1995.
- [Schw96] *Systematic Hypermedia Application Design with OOHDM*, D.Schwabe, G.Rossi, S.Barbosa, Proceedings ACM-Hypertext'96, 1996.

AACE CONFERENCE

Call for Participation

Hosted by Universiti Malaysia Sarawak



**International Conference on Computers in Education**  
December 2-6, 1997  
Kuching, Sarawak, Malaysia