



Escuela Superior de Informática

Universidad de Castilla-La Mancha

Ingeniería web dirigida por modelos

Ingeniería del Software II

2009/2010

Ángel Escribano Santamarina

ÍNDICE

INTRODUCCIÓN	3
CONCEPTOS.....	3
METODOLOGÍAS	5
HERRAMIENTAS Y PLATAFORMAS	9
EJEMPLO	11
CONCLUSIONES	16
REFERENCIAS	17

1. INTRODUCCIÓN

Los problemas encontrados en la construcción de aplicaciones web surgen a partir de la rápida expansión de Internet y el avance en tecnologías web.

Además, las actividades comerciales en la red están aumentando vertiginosamente, por lo que el alcance y la complejidad de las aplicaciones están creciendo cada vez más. A esto se le añade la migración de aplicaciones de escritorio a entornos web.

La evolución de las aplicaciones web también implica el aumento de la complejidad en diseñar, desarrollar, mantener, y gestionar estos sistemas de información.

Debido a esto, los sistemas son construidos en periodos temporales muy cortos, sin el apoyo de herramientas de trabajo adecuadas y utilizando soluciones ad-hoc, lo que lleva a construir aplicaciones web de baja calidad.

A raíz de esto se han ido creando enfoques disciplinados, sistemáticos y metodologías donde se tienen en cuenta aspectos específicos de este nuevo medio.

La ingeniería web surge en este contexto.

2. CONCEPTOS

La ingeniería web es una nueva área de la ingeniería del software que abarca procesos, técnicas y modelos orientados a los entornos Web. Consiste en la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones web de alta calidad.

Los principales ámbitos de la ingeniería web incluyen, entre otros, los siguientes aspectos:

- *Diseño de procesos de negocio para aplicaciones web.*
- *Herramientas CASE para aplicaciones web.*
- *Generación de código para aplicaciones web.*
- *Desarrollo web colaborativo.*
- *Modelado conceptual de aplicaciones web.*

- *Diseño de Modelos de datos para sistemas de información web.*
- *Entornos de desarrollo de aplicaciones web integrados.*
- *Herramientas de autor para contenido multimedia.*
- *Pruebas de rendimiento de aplicaciones basadas en web.*
- *Personalización y adaptación de aplicaciones web.*
- *Modelado de procesos para aplicaciones web.*
- *Herramientas y métodos de prototipado.*
- *Control de calidad y pruebas de sistemas.*
- *Ingeniería de requisitos para aplicaciones web.*
- *Aplicaciones para la Web Semántica.*
- *Factorías de software para la web.*
- *Métodos, herramientas y automatización de pruebas para aplicaciones web.*
- *Aplicaciones web móviles y ubícuas.*
- *Usabilidad de aplicaciones web.*
- *Accesibilidad para la web.*
- *Metodologías de diseño web.*
- *Diseño de interfaces de usuario.*
- *Métricas para la web, estimación de costes y medición.*
- *Gestión de proyectos web y gestión de riesgos.*
- *Desarrollo y despliegue de servicios web.*

En los últimos años existe una tendencia importante a reconocer que las aplicaciones basadas en el Web deben expresarse a un nivel de abstracción mayor que aquél que ofrecen las primitivas propias de las plataformas tecnológicas.

OMG en los últimos años ha reconocido que el enfoque de modelado es una forma potente de especificar sistemas y así lo demuestra su estrategia *Model Driven Architecture* (MDA) con la cual se busca ofrecer al diseñador la posibilidad de expresar la estructura, comportamiento y funcionalidad del sistema independientemente de los aspectos tecnológicos y lograr con ello además la posibilidad de una fácil integración con otros sistemas.

La Ingeniería Web Dirigida por Modelos (MDWE) es la aplicación de la Arquitectura Dirigida por Modelos al campo del desarrollo de aplicaciones web donde puede resultar especialmente útil debido a la evolución continua de las tecnologías y plataformas web.

En esta dirección existen propuestas de modelado de las cuales podemos destacar dos vertientes importantes:

- Metodologías orientadas al diseño navegacional cuyo objetivo es construir aplicaciones hipermedia en sistemas estáticos. La mayoría de estas aproximaciones están basadas en el Modelo Relacional clásico, o bien en extensiones de éste. Algunos ejemplos destacables de estas iniciativas son OOHDM, WebML, ADM, AutoWeb y RMM.
- El otro grupo de aproximaciones se basan en la idea de extender los métodos de desarrollo orientados a aplicaciones dinámicas tratando de introducir la semántica de la hipermedia como característica inherente a este nuevo tipo de sistemas software. Este tipo de aproximaciones de introducir características navegacionales al modelo OO. En este grupo podemos encontrar los métodos UWE, WSDM, EORM, OOW y OO-Method.

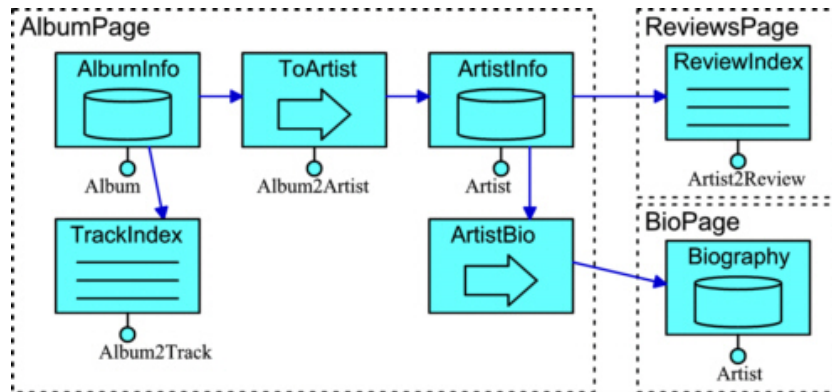
3. METODOLOGÍAS

A continuación se comentan algunas de las metodologías enumeradas en el apartado anterior:

- **UWE (UML-based Web Engineering)** sirve para modelar aplicaciones web, y presta una especial atención a la sistematización y personalización (sistemas adaptativos). Provee de perfiles UML, metamodelos, un proceso de desarrollo dirigido por modelos, y herramientas de soporte para el diseño sistemático de aplicaciones web (ArgoUWE y MagicUWE). Utiliza notación basada en UML 2.0 (OMG): para aplicaciones Web en general y para aplicaciones adaptativas en particular. La metodología consta de seis modelos:
 - *Modelo de casos de uso para capturar los requisitos del sistema.*
 - *Modelo conceptual para el contenido (modelo del dominio).*
 - *Modelo de usuario: modelo de navegación que incluye modelos estáticos y dinámicos.*
 - *Modelo de estructura de presentación, modelo de flujo de presentación.*
 - *Modelo abstracto de interfaz de usuario y modelo de ciclo de vida del objeto.*
 - *Modelo de adaptación.*

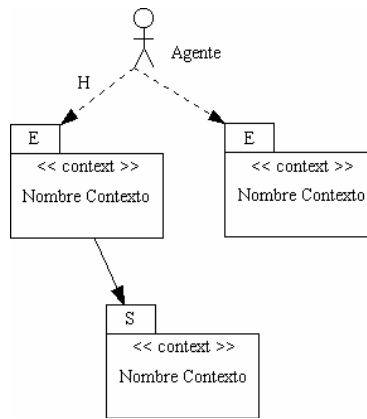
- **Web Services Distributed Management (WSDM)**, se pronuncia *wisdom*) es una especificación basada en servicios web para gestionar y monitorizar el estado de otros servicios. Es un estándar OASIS (Organization for the Advancement of Structured Information Standards), y WSDM consiste en dos especificaciones:
 - Management Using Web Services (MUWS): define como representar y como acceder a las interfaces de gestión de recursos expuestos como servicios web. Define un conjunto básico de operaciones de gestión sobre los servicios, tales como identificación, métricas, configuración y relaciones, además de un formato de eventos estándar.
 - Management Of Web Services (MOWS): define como manejar servicios web como recursos y como describir y acceder a las capacidades de gestión utilizando MUWS. Esta especificación permite a las aplicaciones de gestión de servicios web interoperar entre sí.

- **WebML (Web Modeling Language)**: Es una metodología de modelado visual de aplicaciones web, centrada especialmente en las aplicaciones de uso intensivo de datos, separando el contenido de la información en páginas, navegación y presentación, que se pueden definir y desarrollar de forma independiente. Permite la especificación de operaciones de manipulación de datos para actualizar la aplicación.
 Cuenta con cuatro perspectivas: el Modelo Estructural (de los datos de la aplicación), el Modelo de Hipertexto (para cada hipertexto describe qué páginas lo componen, y cómo navegan), el Modelo de Presentación (disposición y apariencia gráfica), y el Modelo de Personalización (para definir operaciones específicas para usuarios ó grupos de usuarios, ya que se almacenan como entidades en el Modelo Estructural). Dispone de una herramienta CASE (WebRatio).



Ejemplo de la especificación de composición y navegación de un hipertexto

- **OOWS:** Es una extensión del método OO-Method (ya basado en modelos), al cual se le añaden a las técnicas de modelado conceptual, capacidades de expresar la hipermedia inherente a las aplicaciones web. Consta de cinco modelos, que se especifican a continuación:
 - El Modelo de Objetos define la estructura y las relaciones estáticas entre clases identificadas en el dominio del problema.
 - En el Modelo Dinámico se describen las posibles secuencias de servicios y los aspectos relacionados con la interacción entre objetos.
 - El Modelo Funcional captura la semántica asociada a los cambios de estado entre los objetos motivados por la ocurrencia de eventos o servicios.
 - El Modelo de Navegación define la semántica navegacional asociada a las clases de los objetos del modelo. Es en este modelo donde se explicita la navegación permitida en la aplicación para cada agente del sistema, por lo que se realiza un mapa navegacional por cada uno, en el que se definen nodos (posibles puntos de interacción con el usuario) y arcos (posibilidad de alcanzar otro nodo). A continuación se muestra un ejemplo:



Mapa de navegación para un agente

- El Modelo de Presentación captura los requisitos básicos de presentación de información. Está fuertemente basado en el modelo de navegación y permite definir, de una manera abstracta, la estructura lógica de presentación de los objetos navegacionales en la interfaz de usuario.

Esta metodología también cuenta con un entorno de desarrollo para aplicaciones web (OOWS Suite).

- **NDT (Navigational Development Techniques):** Es una metodología orientada a la Ingeniería Dirigida por Modelos (MDE). Aunque en un principio se centraba en las fases de ingeniería de requisitos y análisis, se ha ido ampliando a otras fases del ciclo de vida. Define un conjunto de metamodelos para las fases de requisitos y análisis, y una serie de transformaciones y reglas que permiten obtener los modelos de análisis a partir de ellos. Los metamodelos se representan a partir de MOF (Meta-Object Facility), mientras que las transformaciones se definen mediante QVT (Query/View/Transformation). Consta de una herramienta denominada NDT Suite.

Todas estas metodologías necesitan transformar los modelos para generar aplicaciones web. Según el framework MDA, las transformaciones se pueden aplicar para establecer un proceso de desarrollo trazable desde los modelos

abstractos (CIM, PIM) a los modelos dependientes de la plataforma, ó incluso directamente a su implementación.

La mayoría de las metodologías utilizan herramientas CASE para realizar estas transformaciones. Estas herramientas se basan en técnicas de generación de código para obtener aplicaciones web a partir de un reducido conjunto de modelos conceptuales ó de diseño.

Las transformaciones se pueden dividir en Transformaciones Modelo-a-Modelo, y Transformaciones Modelo-a-Código.

Las Transformaciones **Modelo-a-Modelo** se pueden separar en:

- Transformaciones verticales que convierten modelos de un mayor nivel de abstracción en otros de un nivel menor.
- Transformaciones horizontales que describen el mapeo entre modelos del mismo nivel de abstracción.

Las transformaciones verticales utilizan lenguajes como QVT, ATL ó AGG. A veces incluso se definen como mecanismos de mezcla para introducir nuevos conceptos como estilos de arquitectura, requisitos de usuario, y medida de la calidad.

Las transformaciones horizontales se utilizan para mantener la consistencia de las especificaciones de los modelos, comprobando que estos modelos no imponen requisitos contradictorios en sus elementos comunes.

Las transformaciones **Modelo-a-Código** se llevan realizando más tiempo, aunque a menudo se han utilizado lenguajes de propósito general (C++, Java ó Python). Aunque un nuevo estándar del OMG ha establecido las características propias de los lenguajes de transformación Modelo-a-Código, y algunas herramientas han sido adaptadas para incluirlas.

4. HERRAMIENTAS Y PLATAFORMAS

Las metodologías comentadas anteriormente utilizan distintas herramientas para llevar a cabo diversas operaciones:

- ArgoUWE: Es una extensión de la herramienta de código abierto ArgoUML a la cual se le han añadido capacidad de modelado, navegación y estructuras de presentación.
- WebTE: Es una herramienta de UML que soporta XMI permitiendo la introducción de los modelos y transformaciones en un motor de transformación que los ejecuta y produce una aplicación web en JavaEE.
- WebRatio: Es una herramienta comercial que da soporte a la metodología WebML. Trabaja con metamodelos basados en MOF, modelando los objetos de negocio mediante los estándares UML ó E/R mientras que el front-end se modela con WebML. A partir de aquí, para esos modelos se genera automáticamente la aplicación en la arquitectura JavaEE y las fuentes de datos SQL/XML.
- Eclipse Modelling Project (EMP): Proporciona una gran cantidad de facilidades de desarrollo dirigido por modelos de gran calidad, entre otros:
 - Un framework de modelado común denominado EMF
 - Meta-editores como GMF (Graphical Modelling Framework)
 - Motores de transformación como ATL o VIATRA, entre otros
 - Generadores de código como MOFScript
- NDT Suite: Es un conjunto de herramientas para aplicar la metodología NDT en entornos prácticos. Consta de las siguientes:
 - NDT-Profile es un perfil (*profile*) definido en la herramienta Enterprise Architect. Este perfil aporta diversas herramientas y artefactos propios de NDT.
 - NDT-Driver permite, a partir de un fichero de NDT-Profile como entrada, ejecutar las transformaciones automáticamente, y a partir de los requisitos y los modelos obtiene prototipos HTML automáticos.
 - NDT-Quality toma como entrada un fichero de Enterprise Architect y comprueba que la trazabilidad y las normas de NDT se cumplen.
- OOWS Suite: Es un entorno de desarrollo que da soporte a la metodología OOWS. Se integra con la herramienta comercial OlivaNOVA, que implementa el proceso de desarrollo de O-Method. El entorno consta de un editor gráfico denominado OOWS Visual Editor, el cual asocia a cada primitiva conceptual su representación gráfica o textual.

Otras herramientas CASE pueden ser:

AriadneTool, el framework DaVinci, MIDAS-CASE, VisualWade, M2DAT.

5. EJEMPLO

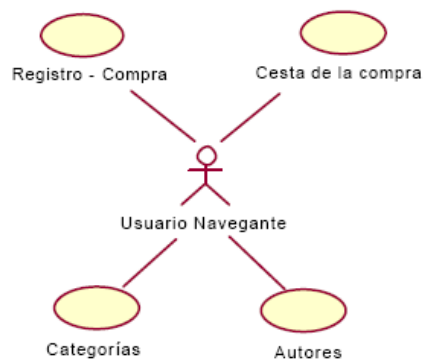
A continuación se muestra el ejemplo del desarrollo de una aplicación web para la compra de discos por internet:

“Existen dos tipos de usuarios, los Administradores y los Usuarios Navegantes. Mientras que los primeros se encargan de la gestión de los álbumes a la venta, los segundos son usuarios comunes (compradores).

Las compras que realicen los Usuarios Navegantes se deberán ir incluyendo, simbólicamente, en una cesta de la compra; el usuario podrá consultar en cualquier momento el contenido de su cesta y realizar modificaciones sobre su contenido. Esta cesta de la compra se creará en el momento en el que se reciba la petición de entrada en el sistema y pertenecerá al usuario que está navegando en ese momento; todas las operaciones que el usuario realice sobre el sistema se harán de forma anónima, de modo que el usuario no deberá identificarse (registrarse) hasta que no vaya a confirmar su compra; para comprar un álbum se deberá llegar a él a través del autor o de la categoría a la que pertenece; desde la página de inicio podremos acceder a un listado de autores o a un listado de categorías y desde ahí al listado de los álbumes del autor o de la categoría que hayamos seleccionado; cuando seleccionemos un álbum de la lista, se mostrarán todos los datos de ese álbum y se podrá comprar. Esto hará que el álbum sea incluido en la cesta de la compra de ese usuario y se muestre su contenido actual; mientras veamos el contenido de la cesta, podremos cambiar el número de unidades que se desea adquirir de cada álbum de los comprados hasta el momento o eliminar alguna de las compras de la cesta; cuando se decida confirmar la compra se realizarán dos acciones: la primera consistirá en crear una factura (para lo que el comprador debe haberse identificado) y la segunda será reducir el stock de los álbumes comprados; cuando se haya confirmado una compra, ya no se podrá modificar el contenido de la cesta.

Los administradores se encargan de gestionar y mantener el catálogo de productos del sistema, así como mantener la lista de autores que poseen discos en la tienda”.

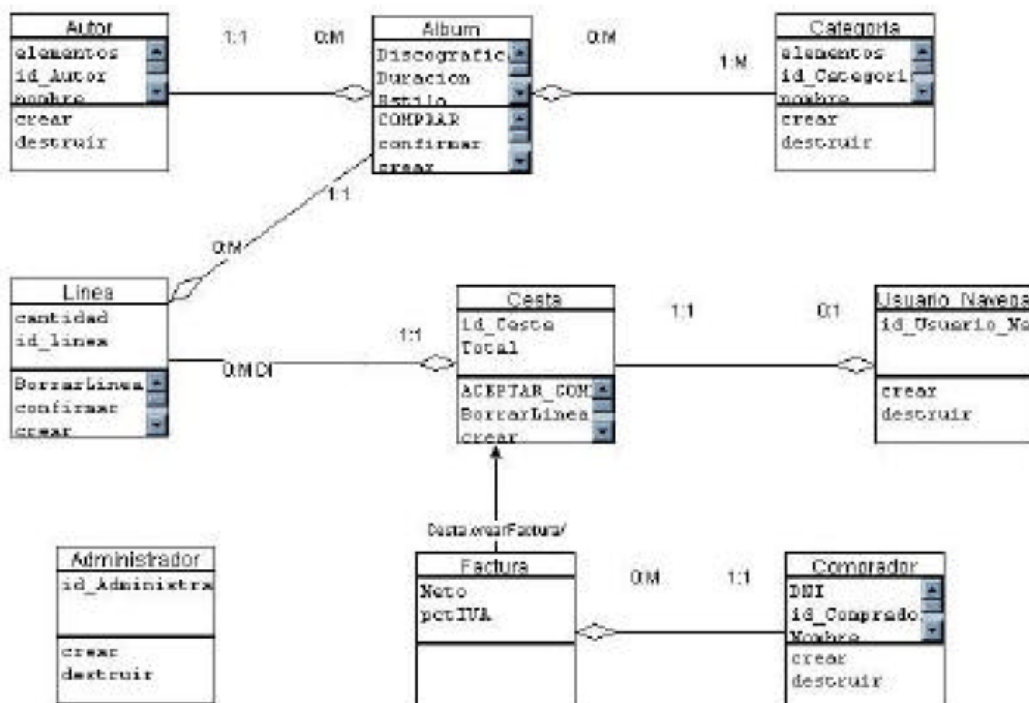
El primer paso es la fase de la **Especificación del Problema**, lo cual genera diagramas de casos de uso a partir de la funcionalidad encontrada en el análisis de los requisitos. A continuación se muestra el caso de uso para el agente Usuario Navegante.



Caso de uso del agente Usuario Navegante

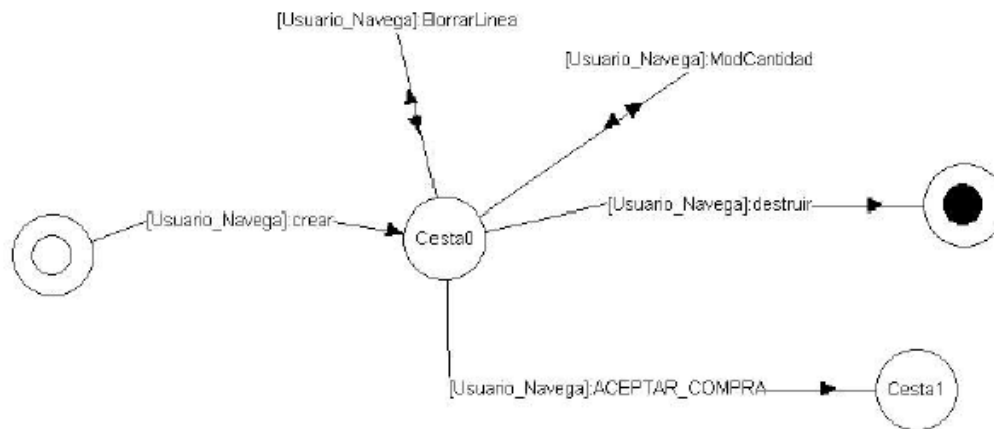
En la fase de modelado conceptual se construyen los siguientes modelos: Modelo de Objetos, Modelo Dinámico, Modelo Funcional, Modelo Navegacional y Modelo de Presentación.

A continuación se muestra el Modelo de Objetos, para realizar el cual se han estudiado los requisitos funcionales del sistema desde un punto de vista OO, asociando la funcionalidad a cada usuario según los casos de uso mostrados en la fase anterior.



Modelo de Objetos

Después del Modelo de Objetos, se construye el Modelo Dinámico donde se describen las vidas válidas de los objetos representando el comportamiento de cada clase del sistema. A continuación se muestra la parte del Modelo Dinámico para la clase Cesta.



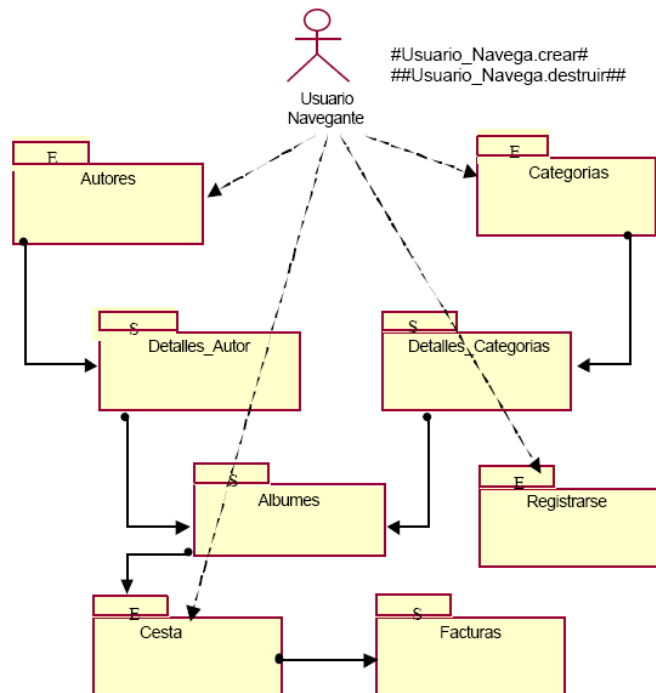
Parte del Modelo Dinámico para la clase Cesta

El siguiente paso es el Modelo Funcional, el cual captura la semántica asociada a los cambios de estado de los objetos. Cada atributo ve modificado su valor dependiendo de la acción que activó el cambio de estado, de los argumentos del evento, y del estado actual de ese objeto. La siguiente figura muestra una parte del modelo para la clase Linea, que contiene la evaluación siguiente:

<i>Atributo:</i> Cantidad	
<i>Categoría:</i> De Estado	<i>Evento:</i> ModCantidad(p_nuevaCantidad)
<i>Efecto:</i> = p_nuevaCantidad	<i>Condición:</i> p_nuevaCantidad > 0

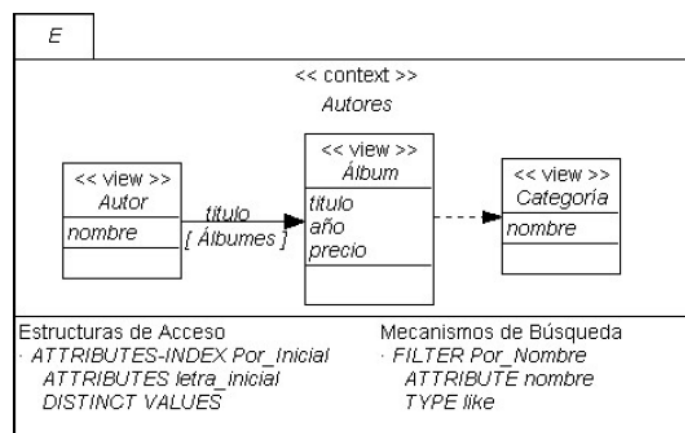
Evaluación del Modelo Funcional para la clase Linea

El Modelo de Navegación es el siguiente paso, donde se estructura el acceso de cada usuario al sistema. A continuación se muestra el mapa de navegación del agente Usuario Navegante, con los contextos de navegación definidos, y los servicios que se ejecutan al iniciar y finalizar sesión. Los contextos de exploración (etiquetados como E) son los que tendrá disponible siempre, y a partir de estos podría alcanzar los demás navegando por diferentes caminos.



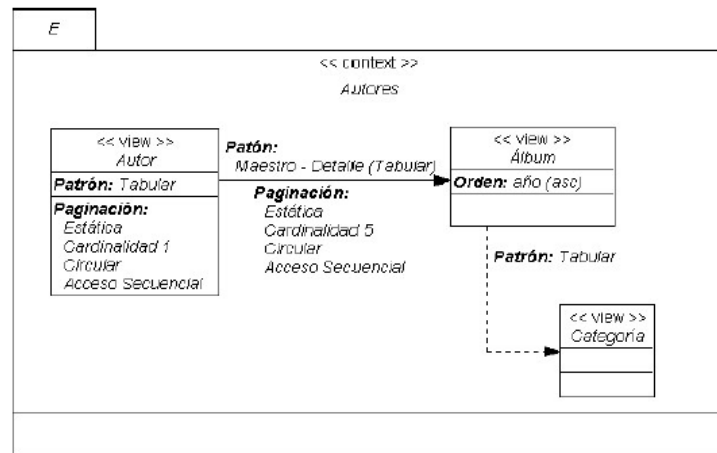
Mapa de Navegación del agente Usuario Navegante

Viendo más en detalle el contexto Autores, se ve que se recupera la información sobre un autor (su nombre), sus álbumes que están disponibles (título, año y precio) y el nombre de la categoría del álbum. Seleccionando el título de un álbum se puede navegar al contexto Álbumes, donde se podrán ver detalles, y añadirlo a la cesta. También se han definido una estructura de acceso que permitirá acceder a los autores por su letra inicial, además de un filtro de tipo aproximado (*like*) para facilitar la búsqueda por nombre.



Contexto Autores del Mapa de Navegación anterior

Por último, se construye el Modelo de Presentación donde se definen los requisitos de presentación de la información para cada contexto del Mapa de Navegación. Por ejemplo, seguidamente se muestra la plantilla de presentación asociada al contexto Autores.



Plantilla de presentación del contexto Autores

Después de la fase de construcción de los modelos, es necesario afrontar la segunda fase, **Desarrollo de la Solución**, donde utilizando una estrategia de compilación de modelos, se obtiene el prototipo software completo de manera automática. A continuación se muestra una posible interfaz de usuario que cumple los requisitos de navegabilidad y de interfaz del agente Usuario Navegante.



Página generada a partir del contexto Autores

6. CONCLUSIONES

La Ingeniería Web Dirigida por Modelos (MDWE) es una disciplina que aplica el paradigma MDA al diseño, desarrollo y mantenimiento de aplicaciones web de forma independiente de la plataforma, lo que tiene una gran serie de ventajas:

- Poder desplegar la aplicación independientemente de la plataforma de destino.
- Poder realizar de forma automática prototipos para evaluar los requisitos, o incluso realizar pruebas de Aseguramiento de la Calidad (QA) sobre los modelos definidos.

Aunque la MDWE proporciona una gran lista de ventajas, todavía es una disciplina bastante joven, debiendo adaptarse a dificultades de diferentes tipos.

Uno de los principales problemas que surge de la aplicación de metodologías dirigidas por modelos tradicionales al desarrollo web es la definición de la navegabilidad de las aplicaciones web, el cual ha sido afrontado y resuelto de manera eficaz. La navegabilidad se está convirtiendo en un aspecto crítico del proceso de desarrollo.

Aún existen problemas que deben tratarse, como son la implementación de requisitos no funcionales como la seguridad o el rendimiento de las aplicaciones web, los cuales no son tenidos en cuenta en la mayoría de las metodologías.

Además, esta nueva área debe adaptarse a la evolución continua de los requisitos de los sistemas web, y a la cantidad de tecnologías y plataformas que surgen.

Las aplicaciones orientadas a servicios y la Web 2.0 proporcionan una infraestructura para integrar múltiples servicios software bajo una interfaz de usuario. RIA, REST o los Servicios Web XML permiten integrar las aplicaciones web actuales con servicios, portales y sistemas de terceros.

7. REFERENCIAS

- [1] <http://www.omg.org/mda/>
- [2] <http://www.omg.org/cgi-bin/doc?omg/03-06-01>
- [3] <http://www.model-transformation.org/>
- [4] <http://planet-mde.org/>
- [5] http://es.wikipedia.org/wiki/Ingenier%C3%ADa_web
- [6] <http://www.andromda.org/>
- [7] <http://www.webml.org/>
- [8] Web Modeling Language (WebML): a modeling language for designing Web sites: <http://www9.org/w9cdrom/177/177.html>
- [9] <http://www.iwt2.org/>
- [10] M.J. Escalona, et al. "Practical Experience in Web Engineering", Springer, 2007
- [11] J. L. Ramírez Terry, "Model Driven Architecture: Ingeniería Informática Empresarial".
- [12] <http://es.wikipedia.org/wiki/OOHDM>
- [13] <http://www.oasis-open.org/>
- [14] <http://es.wikipedia.org/wiki/WSDM>
- [15] <http://es.wikipedia.org/wiki/ArgoUML>
- [16] <http://www.webratio.com/>
- [17] R. Quintero, V. Pelechano, J. Fons, O. Pastor, "Aplicación de MDA al Desarrollo de Aplicaciones Web en OOWS". 2003.
- [18] F. Valverde, P. Valderas, J. Fons, "OOWS Suite: Un Entorno de desarrollo para Aplicaciones Web basado en MDA". IDEAS 2007: X Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software.

[19] Nora Koch et al, “Model-Driven Web Engineering”. The European Journal for the Informatics Professional. April 2008.

[20] M. J. Escalona, “El uso de la ingeniería guiada por modelos para el aseguramiento de la calidad”. Universidad de Sevilla.

[21] O. Pastor, “*OOWS : Una Aproximación para el Modelado Conceptual de Aplicaciones Web*”. 2nd Ibero American Conference on Web Engineering. September 2002. Sta. Fe, Argentina.

[22] J. Fons, O. Pastor, P. Valderas, M. Ruiz. “OOWS: Un Método de Producción de Software en Ambientes Web”. Universidad Politécnica de Valencia

.