

Aplicaciones Orientadas a Objetos con funcionalidad hipermedial en la Web

Ramiro González Maciel, Alejandra Garrido
LIFIA

Facultad de Ciencias Exactas, Universidad Nacional de La Plata.
C.C. 11, (1900) La Plata, Buenos Aires, Argentina.
E-mail: [ramiro,garrido]@sol.info.unlp.edu.ar

ABSTRACT

El framework OO-Navigator brinda facilidades para diseñar e implementar aplicaciones hipermediales que se derivan de agregar funcionalidad de navegación a aplicaciones orientadas a objetos convencionales. Este framework puede ser usado como una herramienta para investigar y experimentar con distintas formas de navegación sobre una hipermedia de este tipo. En su estado actual, el framework solo permite crear links entre nodos de su espacio, por lo que es en cierta medida un sistema cerrado.

En este trabajo se presenta una extensión al framework OO-Navigator que permite su uso en la creación de sistemas de hipermedia que se inserten en la World Wide Web. Esta extensión se ve motivada por la necesidad de analizar como se ven afectados distintos aspectos de una hipermedia (modelo de navegación, percepción del usuario de los límites del sistema, manejo de historia) cuando ésta se inserta en un espacio abierto, de manera que contenga links cuyos destinos sean nodos que no pertenecen a su espacio. Un ejemplo de esta hipermedia de destino es la Web, que en la práctica es infinita y conceptualmente está formada por una agregación de hipermedias.

Con el objetivo de disponer de una herramienta para hacer este tipo de análisis, este trabajo propone una extensión al framework. La extensión a realizar sobre OO-Navigator abarca modificaciones al diseño e implementación en tres áreas. Primero se debe agregar soporte para el modelado de links cuyos destinos estén fuera del modelo de objetos subyacente. Segundo, se deben agregar los mecanismos necesarios para que las interfaces de los nodos sean vistas en un browser de Web. Por último, se debe extender el sistema de historia y registro de links y nodos para que incluyan los nodos visitados externos al sistema. En este trabajo se presentan estas modificaciones al diseño del framework.

INTRODUCCIÓN

A pesar que la investigación en el campo de hipermedia comenzó en los '60s [Elgelbart68], en los últimos años se ha hecho realmente popular entre los usuarios a partir de la creación de la World Wide Web [Berners-Lee92-1] [Berners-Lee92-2]. Debido a esta popularidad y a diversas características técnicas como la independencia de plataforma y la disponibilidad de formatos standard, la Web está siendo considerada por distintas organizaciones como una infraestructura conveniente para desarrollar aplicaciones [Bieber97]. Sin embargo, y en gran parte debido a la poca experiencia y las constantes extensiones que se crean, los diseños de estas aplicaciones corren el riesgo de dejar de lado una de las características más importantes de la Web, el soporte para hipermedia.

El advenimiento de la Web abre la posibilidad a la publicación de interfaces de

un nuevo tipo de aplicaciones : aquellas que combinan navegación con el propio comportamiento. Ante esto hay dos aproximaciones existentes : una que reduce la aplicación a una base de datos sobre la que se realizan queries, y otra que utiliza la técnica de incorporar llamadas a programas desde una interface escrita en HTML, mezclando así las características propias al comportamiento de la aplicación, las de navegación y las de interface a un mismo nivel (con la consiguiente pérdida en reuso y mantenibilidad).

El soporte de hipermedia permite a los usuarios navegar por el espacio de información de un sistema, siguiendo caminos y relaciones previamente definidos, y también definir nuevas conexiones entre elementos de información. Existen dos formas de disponer de soporte para hipermedia en un sistema de información. Una es crear este sistema a partir de elementos de hipermedia. La otra, conocida como "funcionalidad de hipermedia", consiste en agregar esos elementos en un nivel distinto al de los elementos del modelo de información.

FUNCIONALIDAD DE HIPERMEDIA EN APLICACIONES OO

El concepto de funcionalidad de hipermedia consiste en la facilidad de agregar características de hipermedia a distintos sistemas de información, de manera tal de proveer a sus usuarios de un mecanismo que, de forma asociativa, permita acceder, analizar y organizar la información del sistema [Oinas95].

La aproximación de la funcionalidad de hipermedia conserva las relaciones que definen la semántica y el comportamiento de una aplicación. El agregado de esta funcionalidad permite que los usuarios de estas aplicaciones agreguen anotaciones y relaciones entre elementos del sistema. Estas características son beneficiosas en distintos tipos de sistemas de información. Un ejemplo concreto de como se beneficia una aplicación al agregarle esta funcionalidad es el caso de un ambiente CASE orientado a objetos. Estos ambientes permiten manejar las entidades que se usan durante el diseño, como clases, objetos, subsistemas, especificar sus características desde distintos puntos de vista, y verlas en distintas notaciones y gráficos. Una herramienta CASE puede verse beneficiada por el agregado de funcionalidad de hipermedia en los siguientes puntos:

- ⇒ un grupo de desarrolladores pueden trabajar sobre el mismo diseño, donde cada uno puede agregar sus propias anotaciones y definir nuevas relaciones entre distintos elementos. Algunas de estas se podrían compartir, de manera que estén accesibles para los otros diseñadores
- ⇒ la posibilidad de definir relaciones entre los elementos, y "bookmarks" permite a cada usuario adaptar la interface a su conveniencia.
- ⇒ en un proceso iterativo como lo es el diseño orientado a objetos, la posibilidad de tener acceso a la historia de navegación facilita este proceso, de manera que el diseñador sabe en todo momento "como llegó a un lugar"
- ⇒ cada diseñador puede definir un tour guiado de manera que los otros diseñadores puedan seguir de forma ordenada las modificaciones hechas a un diseño por el primero.

En este artículo se analizan las modificaciones que se deben hacer a un framework orientado a objetos que provee funcionalidad de hipermedia a aplicaciones, de manera de poder integrarlo a la Web. El framework sobre el que se desarrolla este trabajo se describe en la sección siguiente.

OO-NAVIGATOR

En esta sección hacemos una breve descripción del framework OO-Navigator,

comentando solo los aspectos relevantes a este trabajo. Una descripción completa del framework se puede encontrar en [Garrido97].

El objetivo del framework OO-Navigator es proveer una arquitectura que permite agregar funcionalidad de hipermedia a una aplicación construida usando el paradigma de orientación a objetos.

OO-Navigator define una arquitectura en capas, en las que se separa el modelo de objetos de la aplicación de los aspectos propios de la funcionalidad de hipermedia agregados. De esta manera se consigue una de las principales virtudes de este framework , que es facilitar esta integración sin mezclar u obscurecer con esta nueva funcionalidad el modelo de la aplicación.

Las tres capas que define OO-Navigator son :

Nivel de objetos : en este nivel se encuentran los objetos que forman el modelo de la aplicación

Nivel de hipermedia : en esta capa se ubican los componentes definidos por el framework. Estos componentes modelan los conceptos que definen una hipermedia

Nivel visual : en este último nivel se sitúan los aspectos de la interface o apariencia visual de los componentes de la hipermedia.

También están definidos en OO-Navigator la forma en que se relacionan los componentes de las distintas capas.

En el nivel de hipermedia se definen los componentes que modelan conceptos hipermediales. Los relevantes a este trabajo son :

Hipermedia : es una red de nodos y links que brinda la posibilidad de navegar entre ellos

Nodo : es un elemento de información autocontenido

Link : representa la relación entre dos o más nodos. La característica fundamental de los links es que a través de ellos se puede navegar de un nodo a otro. Cada link puede ser accedido desde un ancla en el nodo de origen.

Ancla de link : es la parte del nodo de origen donde puede ser activado el link.

Representación de nodo : define en términos de elementos de interface como se presentará un nodo. Un nodo puede tener varias representaciones asociadas.

UNA ARQUITECTURA QUE INTEGRA OO-NAVIGATOR Y LA WEB

De acuerdo con las características del framework, nuestra extensión debe cumplir con dos objetivos principales ; el primero es facilitar el uso de un browser de Web para mostrar los distintos aspectos de la interface de una hipermedia, lo que incluye la representación de los nodos, la historia, información sobre los contextos. El segundo es permitir que las hipermedias creadas con el framework tengan links cuyos destinos sean nodos de la Web. En esta sección desarrollaremos una descripción de la primera extensión, desde el punto de vista de la arquitectura del sistema. La sección "Modificaciones al framework" analiza la segunda parte de la extensión.

La nueva arquitectura a usar define dos componentes principales, por un lado el cliente, que es el usado por el usuario para navegar la hipermedia, y el servidor, que es donde se almacena la hipermedia y quien lleva el control de la navegación. Un esquema de la arquitectura se encuentra en la figura 1.

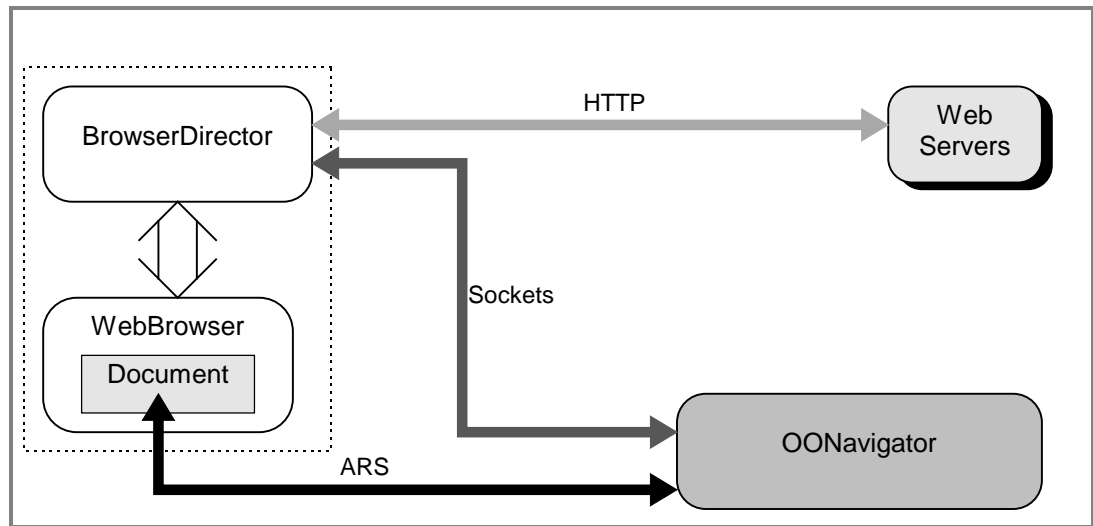


Figura 1 : Esquema de la arquitectura

ARQUITECTURA DEL CLIENTE Y SERVIDOR

El objetivo de la arquitectura del cliente es proveer una integración entre un browser de Web y el OO-Navigator. El cliente está compuesto por dos elementos principales. Uno es un browser de Web y el otro es un módulo que controla este browser.

La función principal del browser es mostrar las representaciones de los nodos de la hipermedia, y los documentos de la Web por los que navega el usuario.

El director del browser, llamado **BrowserDirector**, lleva un registro de las acciones que efectúa el usuario, informando al servidor de estas. También provee una interface al usuario para manejar distintos aspectos de la hipermedia, como la historia e información sobre los contextos.

El esquema de esta arquitectura cliente es similar a la presentada en [Grønbæk97]

Desde el punto de vista del servidor, las modificaciones consisten en agregar el soporte necesario para manejar los canales de comunicación mencionados en la sección siguiente.

CONEXIÓN ENTRE LOS DOS COMPONENTES

Como se observa en la figura 1, existen tres canales de comunicación que parten del cliente. El primero es el que comunica el Director con el OO-Navigator. Esta comunicación bidireccional permite al Director comunicarle a la hipermedia las distintas acciones de navegación del usuario, y al OO-Navigator indicarle al Director que acciones debe iniciar en el browser. Una de estas acciones es la carga de un documento, acción que se inicia al navegar el usuario un link de la hipermedia que tiene como destino un documento de la Web. En este caso, el Director crea una comunicación con un Web server, requiriendo el documento, y le indica al browser que debe mostrarlo. Este último es un ejemplo del canal de comunicación entre el Director y un Web server.

El último canal de comunicación es el que conecta las interfaces de las representaciones de los nodos como se ven en un browser, con los objetos que las modelan en el OO-Navigator. Este canal de comunicación se explica en la sección 4.6

MODIFICACIONES AL FRAMEWORK

En esta sección se detallan los aspectos de la arquitectura y diseño del framework que debieron ser modificados o extendidos. Las modificaciones abajo descriptas afectan principalmente el nivel de hipermedia. Cada una de estas modificaciones se explica usando un formato compuesto por cinco partes. Este formato está basado en la estructura usada para la definición de patterns [Gamma95]. Las cinco partes que conforman la descripción de una modificación son :

- Título : permite hacer referencia al problema y a su solución.
- Problema : donde se detalla brevemente el problema.
- Contexto : en esta sección se comenta el contexto en el que hay que solucionar el problema, junto con las restricciones que limitan las posibles soluciones.
- Solución : se explica la solución obtenida.
- Consecuencias : donde se detalla de que manera afecta la solución a distintas partes del sistema. Dado que en algunos casos existen distintas formas de implementar la solución propuesta, en este punto también se presentan estas variantes.

Cada modificación se presenta con este formato en una de las subsecciones siguientes. En lo siguiente, cuando hacemos referencia a una hipermedia, nos referimos a las hipermedias creadas con OO-Navigator

NAVEGACIÓN Y SESIÓN

Problema :

La historia de navegación debe incluir a nodos de la hipermedia y a nodos de WWW.

Contexto :

OO-Navigator fue creado para manejar cierto tipo de hipermedias, por lo que impone algunas invariantes a lo largo de su diseño que deberán ser modificadas. La primera de ellas es que el concepto más amplio que maneja el framework es el de hipermedia. De acuerdo al framework una hipermedia tiene dos responsabilidades principales, que consisten en manejar su estructura (nodos, links, contextos) y su historia de navegación. Al integrar OO-Navigator con la Web, es necesario separar estas dos responsabilidades, ya que no todos los nodos por los que navegará el usuario pertenecen a una hipermedia del framework.

Solución :

Por un lado cada hipermedia seguirá haciéndose cargo de los elementos que la componen, pero el manejo de la historia pasará a estar a cargo de un objeto que represente una sesión. Una sesión se encargará de tener un registro de la historia de navegación de un usuario, el cual puede navegar dentro de una hipermedia y también navegar por nodos de Web. La clase del framework `Hypermedia` debe ser modificada para que informe de la activación de sus elementos a la sesión en la que se encuentra activada.

Consecuencias :

Esta modificación implica que una hipermedia podrá estar activa en una sola sesión.

El concepto de sesión será usado también para resolver otros problemas relacionados con la integración del OO-Navigator en la Web. Ver por ejemplo el problema "WebNodes".

WEBNODES

Problema :

Representación dentro del framework de los nodos de WWW.

Contexto :

Para lograr la integración es necesario modelar los nodos de Web por dos motivos, el primero es que pueden ser el destino de un link con origen en la hipermedia, y el segundo es que se quiere tener un registro de todos los nodos por los que navega un usuario en una sesión, independientemente de si pertenecen o no a una hipermedia.

Las características de los nodos de WWW que influyen nuestra decisión son :

- no pertenecen a ninguna hipermedia creada con el framework
- son accesibles desde distintas hipermedias modeladas con OO-Navigator
- es imposible e innecesario tener representados todos los nodos de la WWW en nuestro modelo

Actualmente el framework define una clase `HypermediaComponent` que abstrae a todas las entidades que pueden formar una hipermedia (nodos y links). Todo componente de una hipermedia, y por lo tanto instancia de alguna subclase concreta de `HypermediaComponent`, es capaz de ser navegado, lo que consiste básicamente en activarse mostrando la información que representa, y registrarse, informando a la hipermedia a la que pertenece que ha sido activado. De acuerdo con esto, cada componente está relacionado con su hipermedia. La primera característica de los nodos de Web arriba enumerada impide usar el concepto de `HypermediaComponent` para representar a los de Web.

Solución :

De acuerdo a estas características y nuestros requerimientos, se decidió tomar una representación "lazy" o por demanda, donde solo se modelan los nodos de WWW

cuando es necesario. Cada nodo Web estará modelado por una instancia de la clase `WebNode`. Las responsabilidades de esta clase incluyen ejecutar la secuencia de activación y registración. Para cumplir con estas dos responsabilidades, cada `WebNode` está relacionado con una sesión, a la cual le informa cuando es activado para que sea registrado.

Consecuencias :

La activación de los `WebNodes` se explica en el problema 'Representaciones en un browser'.

En esta solución los `WebNodes` no almacenan el contenido de los nodos, es decir su código HTML. Esto afecta la forma en que se activan los links (ver problema "Anchors en WebNodes")

ANCHORS EN WEBNODES

Problema :

¿Cómo se modelan los links entre `WebNodes` ?

Contexto :

Dado que no es necesario tener el contenido de los nodos en la instancia de `WebNode` que los representa, y que queremos almacenar en la historia la secuencia de nodos visitados, debemos disponer de alguna forma de notificación acerca de cuales son los nodos que se van visitando a partir de un nodo de Web dado.

En los nodos que pertenecen a la hipermedia este problema no se presenta, dado que los links y las anchors se modelan explícitamente con las clases `Link` y `Anchor`.

En el caso de los nodos Web, la activación de los anchors será ejecutada por el browser de Web del usuario

Solución :

El Web browser está controlado por el `BrowserDirector`, el cual es capaz de detectar la activación de los links en los documentos de WWW. Cada vez que el usuario activa un link, se sigue la siguiente secuencia, esquematizada en la figura 2. La notación de es similar a la usada en los diagramas de interacción en [Gamma95]

- 1) El usuario activa un link
- 2) El `BrowserDirector` detecta la activación
- 3) El `BrowserDirector` le informa al objeto que maneja la sesión de este suceso. Este aviso involucra una comunicación a través del canal que conecta al `BrowserDirector` con `OO-Navigator`. Esta comunicación incluye información sobre el destino del link.
- 4) Cuando la sesión (parte de `OO-Navigator`) recibe esta información, se encarga de crear un nuevo `WebNode` que represente el nodo activado por el usuario en el browser
- 5) La sesión activa el `WebNode`.
- 6) Como resultado de esta activación, el nodo se agregará a la historia.

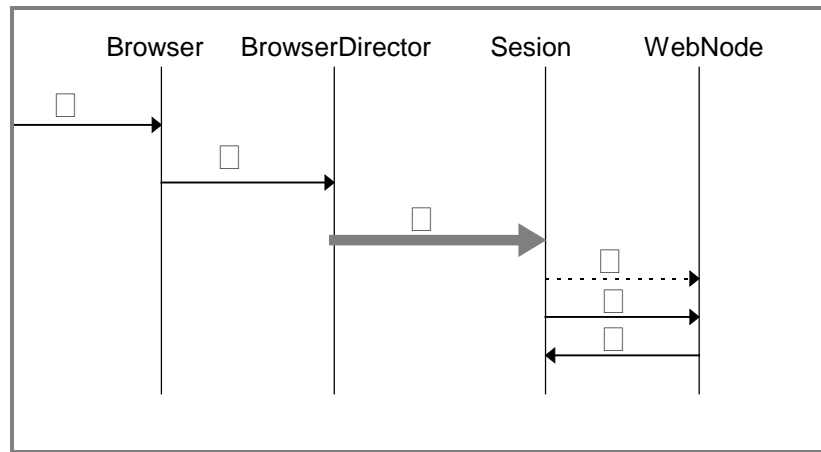


Figura 2

Consecuencias :

Existen dos variantes de esta solución. La primer variante implica que el documento destino es mostrado en el browser y después se inicia la comunicación con el BrowserDirector. La segunda sigue una secuencia inversa.

La primer variante tiene como ventaja que la activación del documento en el browser es más rápida, por lo que el uso es más amigable para el usuario. La segunda permite una validación del documento a cargar.

ENDPOINTS EN WEBNODES

Problema :

¿Cómo se modelan los links cuyo origen se encuentra en la hipermedia y cuyos destinos son WebNodes ?

Contexto :

Cuando un link es activado, es su responsabilidad activar el nodo destino. Existen distintas variantes para la activación de este nodo, ya que el destino puede ser fijo o computado. Por esto OO-Navigator define una jerarquía de objetos que encapsulan la forma de calcular un destino (*EndpointSolvers*), con subclases específicas para cada forma distinta de activación. En la figura 3 se muestra un diagrama de la jerarquía de *EndpointSolver*.

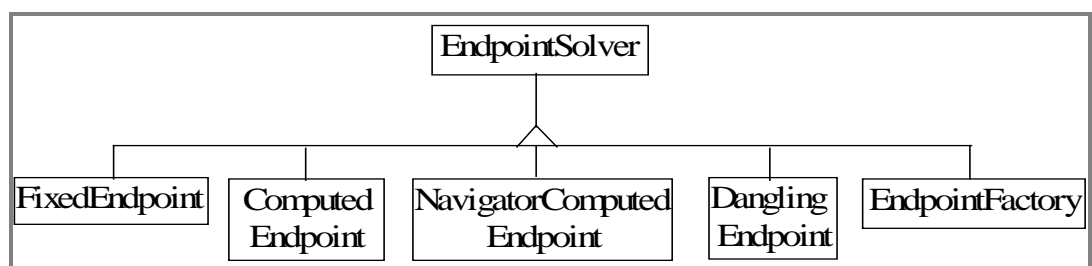


Figura 3

Solución :

En el caso de los WebNodes, usaremos una estrategia de solución en la que el destino sea fijo, dado que cuando se crea un link hacia un documento de WWW, el destino se conoce de antemano.

Por esto, los endpoints en nodos de Web serán modelados con instancias de FixedEndpoint.

Consecuencias :

Dado que los FixedEndpoints conocen directamente al nodo destino, será necesaria la creación de un WebNode para cada link con destino en la WWW. Otra posible solución consiste en no usar FixedEndpoints sino una nueva clase que almacene la especificación del nodo destino (su URL) y que cree el destino cuando es activado.

REPRESENTACIONES EN UN BROWSER

Problema :

¿Cómo se muestra la interface de la hipermedia en el browser del usuario ?

Contexto :

Como fue comentado anteriormente, uno de los aspectos de la integración consiste en permitir que el usuario use un browser de Web para navegar por la hipermedia.

La versión actual de OO-Navigator usa como interface la provista por VisualWorks [ParcPlace95]. De acuerdo a esto, cada vez que un nodo es activado, muestra su representación en una ventana provista por VisualWorks.

Solución :

El framework debe ser modificado para que cada vez que un nodo sea activado, le comunique a la sesión en la que está corriendo que debe mostrar su representación.

En el caso de que el nodo activado sea un WebNode, la sesión se encargará de que el browser cargue el URL correspondiente, a través del BrowserDirector.

En el caso de un nodo de hipermedia, se enviará al browser la especificación de la representación. El formato de la representación se detalla en el problema "Representación de los nodos"

Consecuencias :

Aparte de la representación de los nodos, otros aspectos de la hipermedia deben ser mostrados al usuario, como el contexto por el que está navegando. Estos aspectos pueden ser visualizados en el browser o en la interface del BrowserDirector.

REPRESENTACIONES DE LOS NODOS

Las extensiones comentadas en la sección anterior trataban fundamentalmente de la capa de hipermedia y la arquitectura del framework. En esta sección se explica como se modifica la capa de presentación o de interface.

Problema :

¿Cómo construir las representaciones de los nodos de forma de aprovechar los distintos elementos de interface disponibles ?

Contexto :

Además de las posibilidades al nivel de hipermedia que implica la integración del framework en la Web, el uso de un browser como interface de usuario provee la posibilidad de usar distintos elementos para la creación de las representaciones de los nodos. El factor detonante es la posibilidad de manejar código HTML, ya que a partir de esto se pueden incluir diversa cantidad de elementos multimediales, como gráficos, animaciones, sonido, y distintas técnicas para la programación de las interfaces, como Java, JavaScript, ActiveX, etc.

Es de destacar también que una gran virtud del framework OO-Navigator está dada por la posibilidad de agregar funcionalidad de hipermedia a una aplicación mientras que conserva las interfaces creadas con elementos de VisualWorks. Es a partir de estos elementos de VisualWorks que se crean las interfaces de los nodos, usando un editor de hipermedias provisto por el framework.

Solución :

El framework es extendido para que las representaciones de los nodos estén construidas con elementos de ambos sistemas, HTML y VisualWorks. Esta integración se hace posible por el uso de una herramienta [ARS97] que permite mostrar ventanas de VisualWorks dentro de un documento de HTML. Para realizar la comunicación entre los widgets que forman la ventana en el browser y los objetos que la modelan en el framework, se usa el canal etiquetado "ARS" en la figura 1.

Consecuencias :

A partir de esta solución es posible la creación de distintos editores de interfaces de nodos, así como también posibilita el uso de editores externos.

CONCLUSIONES

Se ha presentado en este trabajo el diseño de una extensión a un framework de funcionalidad de hipermedia que permite su integración en la Web. Estas modificaciones se presentaron principalmente en el nivel de hipermedia, y también en el nivel de interface.

El diseño de las extensiones se vio simplificado gracias a la arquitectura en capas que presenta el framework. Sin embargo algunas restricciones impuestas por el mismo no permiten agregar algunas características sin que sea necesaria una reestructuración a varios niveles. Uno de los aspectos más notables es la dificultad para poder pasar de una estructura monousuario a una multiusuario. Esto posibilitaría el uso en otras áreas como el aprendizaje colaborativo [Fernández97]. Las próximas modificaciones al framework involucrarán este aspecto.

REFERENCIAS

- Garrido97 Alejandra Garrido. *OO-Navigator : Un framework para hipermedia*. Tesis de grado de la Licenciatura en Informática. Universidad

- Nacional de La Plata, 1997
- Gamma95 E. Gamma, R. Helm, R. Johnson and J. Vlissides. *Design Patterns, Elements of Reusable Object Oriented Software* (Addison Wesley, Reading, MA 1995)
- Bieber97 M. Bieber, F. Vitali. Toward support for hypermedia on the World Wide Web. *IEEE Computer*, Enero 1997.
- Grønþæk97 K. Grønþæk, N. Bouvin, L. Sloth. Designing Dexter-based hypermedia services for the World Wide Web, in *Proceedings of the Eighth ACM Conference on Hypertext, Hypertext '97*. Southampton, UK, April 1997, pp. 146-156.
- ARS97 Applied Reasoning Systems Corp. *Classic Blend Withe Paper*. <http://www.arscorp.com>
- Engelbart68 D. Engelbart, W. English. A research center for augmenting human intellect, in *Proceedings of the Fall Joint Computer Conference*, 1968, pp. 395-410.
- Berners-Lee92-1 T. Berners-Lee *et al.* World-Wide Web: An Information Infrastructure for High-Energy Physics, in *Software Engineering, AI and Expert Systems for High Energy*. 1992. La Londe-les-Maures, France
- Berners-Lee92-2 T. Berners-Lee *et al.* *World-Wide Web : The Information Universe*. Electronic Networking : Research, Applications and Policy, 1992. 1(2).
- Oinas95 H. Oinas-Kukkonen. Developing Hypermedia Systems – The Hypermedia Funcionality Approach, In *Proceedings of the Second Basque International Workshop on Information Technology (BIWIT '95)*, keynote paper, July 1995, San Sebastian, Spain, IEEE Computer Society Press, Loas Alamitos, CA, 1995, pp. 2-8.
- Fernández97 A. Fernández. *Collaborative work to make virtual campuses feel real*. ALT Conference, Telford, UK, Septiembre de 1997.
- ParcPlace95 ParcPlace-Digitaltalk. *VisualWorks User's Guide*. 1995