Hacia la separación de enlaces en sistemas web

A. M. Reina Quintero[1] and J. Torres Valderrama[1]

Dpto. Lenguajes y Sistemas Informáticos.
E.T.S. Ingeniería Informática.
Avda. Reina Mercedes, s/n.
41007 Sevilla, España
{reinaqu, jtorres}@lsi.us.es
http://www.lsi.us.es/~reinaqu

Resumen El volumen de negocio que se genera en la red es cada vez mayor. Las aplicaciones web tienen características diferenciadoras con respecto a las aplicaciones tradicionales. Por ello, ha surgido una nueva disciplina conocida como ingeniería web. Este artículo propone un enfoque basado en la separación de conceptos para mejorar la evolución de las aplicaciones web. Esta propuesta se basa en las ideas aplicadas al sistema de evolución de bases de datos SADES, centrándose sobre todo en las relaciones dinámicas, ya que este tipo de relaciones se pueden aplicar muy bien a los enlaces de navegación.

1. Introducción

En los últimos años, gracias al rápido crecimiento de Internet, y a la proliferación de los ordenadores domésticos, los negocios en la red se han convertido en una fuente importante de ingresos para las empresas. Así, muchas de las aplicaciones actuales se proyectan para ser utilizadas en entornos distribuidos con interfaces web. El rápido crecimiento del número de este tipo de aplicaciones, así como el hecho de que presentan una serie de características diferenciadoras con respecto a las aplicaciones tradicionales (su corto periodo de desarrollo o la importancia de la navegación), han provocado la aparición de una nueva disciplina conocida como ingeniería web, que se preocupa de investigar la problemática asociada a este tipo de aplicaciones.

La investigación en ingeniería web se ha centrado, sobre todo, en la definición de metodologías y lenguajes de modelado que se adecúen mejor a las particularidades de las aplicaciones web que las metodologías y lenguajes tradicionales. Además, se ha buscado dotar de un proceso ingenieril al desarrollo de este tipo de proyecto software, ya que se ha pasado en un corto periodo de tiempo de aplicaciones que eran simples repositorios de información a tener que tratar con complejos sistemas distribuidos.

Los autores de estas metodologías y lenguajes de modelado han sido conscientes, sobre todo a partir de OOHDM [14], de las ventajas de la separación de conceptos. Así, la mayoría de las propuestas como UWE [1], OO-H [5], WS-DM [3], e incluso la propia OOHDM, trabajan con modelos separados para los esquemas de navegación, presentación y conceptual.

En este artículo, se realiza un análisis del metamodelo de UWE, y de las clases generadas por ArgoUWE [7], la herramienta CASE asociada a esta metodología. Se observa que estas clases generadas contienen información de las relaciones entre las mismas, y tomando como inspiración las ideas aplicadas en el desarrollo de SADES [12] (un sistema de evolución de bases de datos de objetos personalizable y extensible), se propone aplicar el concepto de relaciones dinámicas ya que mejora la flexibilidad y la mantenibilidad de la aplicaciones generadas.

La distribución del artículo es la siguiente: En primer lugar, en la sección 2 se da una idea somera del metamodelo de UWE [8]. En la sección 3, se describe brevemente el sistema SADES, poniendo énfasis, sobre todo, en las características que en nuestra propuesta extenderemos a los sistemas web. Luego, en la sección 4 se presenta, en primer lugar, un ejemplo sobre el que se trabajará, y en segundo lugar, se mostrará nuestra propuesta para mejorar el ejemplo. Finalmente, se concluirá el artículo y se apuntarán nuestras líneas futuras de investigación.

2. Metamodelo de UWE

En el campo de la ingeniería web han surgido numerosas propuestas de metodologías y lenguajes de modelado (OOHDM [14], UWE [1], OO-H [5], WS-DM [3], ...), teniendo todas ellas como elemento común fundamental el hecho de que proponen modelos separados para especificar los esquemas conceptual, de navegación y de presentación. Debido al gran maremagnum de metodologías y lenguajes de modelado, en [13] se proponía la definición de un lenguaje común de modelado. Koch en [8] va un paso más allá y propone la definición de un metamodelo común para el desarrollo de aplicaciones web como forma de unificar los distintos constructores de modelado de las distintas propuestas de metodologías web. Como primer paso para este metamodelo común, los autores definen un metamodelo para UWE, su propia metodología.

El metamodelo de UWE puede verse como una extensión conservadora del metamodelo de UML, de tal forma que los elementos del metamodelo de UWE heredan las propiedades de al menos un elemento del metamodelo de UML. En la Figura 1, se muestra la estructuración en paquetes del metamodelo de UWE, así como su relación con el metamodelo de UML. Como puede observarse en la misma, en cierto modo, se están separando competencias, ya que se gestionan por separado el núcleo (que en la Figura 1 se representa como paquete Core), la información contextual y la de comportamiento. Además, dentro del propio paquete Core se separan el paquete con los constructores para los modelos conceptuales, el de los constructores del modelo de navegación y el de los constructores del modelo de presentación. Los paquetes que aparecen sombreados en gris son paquetes del metamodelo de UML.

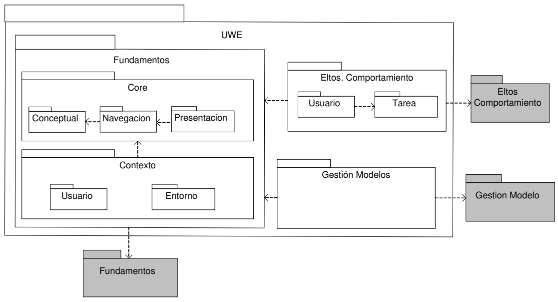

Figura 1. Paquetes del metamodelo de UWE

3. El sistema de evolución de bases de datos de objetos SADES y las relaciones dinámicas

SADES [12] es un sistema de evolución de bases de datos de objetos, construido como una capa situada por encima del sistema de gestión de bases de datos comercial Jasmine. Para desarrollarlo los autores emplearon el principio de separación de conceptos durante todas las etapas de desarrollo. La separación de conceptos se realiza mediante un planteamiento híbrido, en el sentido de que se utilizan diferentes enfoques para separar conceptos. Es decir, para cada concepto o conjunto de conceptos relacionados se estudia cuál es la técnica de separación más apropiada para implementarlo.

Un elemento importante en este tipo de sistemas es la caracterización de las relaciones entre elementos. Hay autores [4] que caracterizan las relaciones entre clases como relaciones estáticas, ya que se fijan en tiempo de compilación, mientras que las relaciones entre instancias las clasifican como dinámicas ya que pueden cambiar en tiempo de ejecución. Rashid en [11], sin embargo, difiere de esta opinión y afirma que las relaciones entre clases no tienen por qué ser fijadas en tiempo de compilación, sino que pueden cambiar en tiempo de ejecución.

Este enfoque de relaciones dinámicas es el que se emplea para implementar los distintos tipos de relaciones existentes en SADES. En la Figura 2 se muestran los dos tipos de relaciones entre los diferentes elementos: relaciones inter-espacios (dibujadas con una línea continua) y relaciones intra-espacios (trazadas con líneas discontínua). Así, una relación se definirá como un constructor semántico y, posteriormente, se transformará directamente en la implementación.

En [12] se demuestra que la información sobre las conexiones entre entidades corta a las propias entidades. Para conseguir la separación de las información de las relaciones y las propias entidades en SADES se elige como mecanismo de separación los filtros de composición [2].

Este artículo propone la extensión del concepto de relaciones dinámicas definido en [11] para ser utilizado en herramientas de desarrollo de sistemas web.

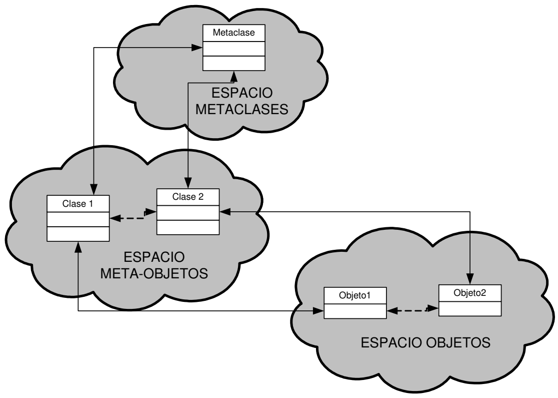

Figura 2. Relaciones entre espacios de SADES

4. Separación de enlaces

Los autores de UWE han implementado una herramienta CASE llamada ArgoUWE [7] para dar soporte a la fase de diseño del proceso de desarrollo UWE. ArgoUWE está implementada como un *plugin* de la herramienta de modelado de código abierto ArgoUML. Para explicar mejor la problemática, hemos modelado un sistema de gestión de publicaciones de grupos de investigación, utilizando la herramienta ArgoUWE, con idea de analizar el código que genera. A partir de ahora nos referiremos a él como ejemplo de publicaciones.

En el apartado 4.1 se presentará el ejemplo de trabajo y se analizarán las clases generadas por ArgoUWE, mientras que en el apartado 4.2, se presentará nuestra propuesta para mejorar esta generación aplicando la idea de relaciones dinámicas.

4.1. Ejemplo y análisis del código generado por ArgoUWE

El modelo conceptual del ejemplo de publicaciones se muestra en la Figura 3. En este diagrama, aparecen las clases estereotipadas, ya que UWE define las clases conceptuales como un estereotipo de la metaclase `Class` del metamodelo de UML, y define una representación gráfica (un círculo en la parte superior derecha) para representar este estereotipo.

El modelo de navegación para el mismo ejemplo se muestra en la Figura 4. Para el modelo de navegación, además de definir un estereotipo para las clases de navegación (véanse los constructores del paquete de navegación en la Figura 5), cuyo símbolo gráfico es un rectángulo en la parte superior derecha de la clase estereotipada, se añaden nuevos constructores, tales como índices, menus, queries y visitas guiadas. En el diagrama del ejemplo se pueden observar un menú (`MenuPPal`), índices (`AuthorByName`, `ItemByYear` e `ItemByYearByKind`), y una estructura tipo query (`SearchByKeyword`).

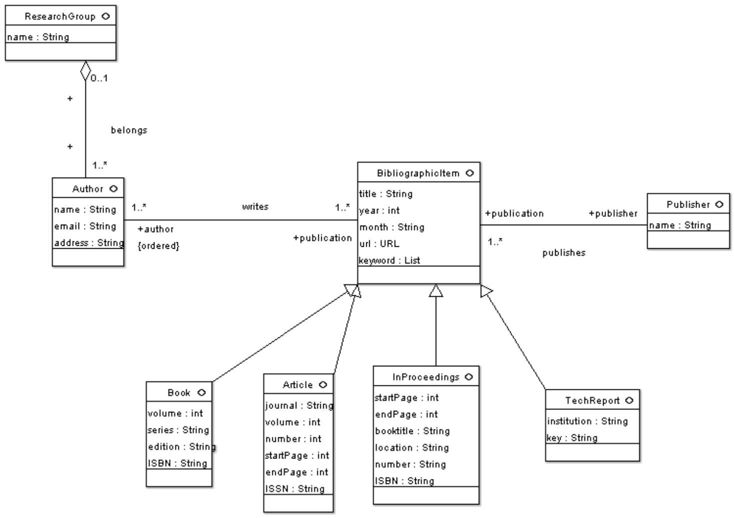

Figura 3. Modelo Conceptual del Sistema de Gestión de Publicaciones de Grupos de Investigación

En la Figura 5, también se puede observar que el enlace se trata como ciudadano de primera clase en el metamodelo. Sin embargo, si analizamos la estructura de las clases que genera la herramienta ArgoUWE, podremos comprobar que la información de la relación entre las clases está embebida en las mismas. Si nos fijamos en la generación de las clases conceptuales, veremos que la clase `BibliographicItem` se genera con un atributo de tipo `Vector` llamado `author`, que contendrá las referencias al conjunto autores de una referencia bibliográfica; mientras que la clase `Author` se genera con un atributo, también de tipo `Vector`, llamado `publication`, que contendrá las referencias a las publicaciones del autor. De la misma forma, las clases generadas a partir del modelo navegacional contienen información de las relaciones entre clases. Así, a partir de la estructura de acceso `MenuPpal`, se genera una clase con tres atributos, un atributo de tipo `AuthorByName`, otro atributo de tipo `SearchByKeyword`, y por último, otro de tipo Vector para almacenar referencias a los items bibliográficos.

4.2. Propuesta de mejora utilizando enlaces dinámicos

El enfoque de relaciones dinámicas presentado en la sección 3 es ideal para los sistemas web, sobre todo debido a que son sistemas que pueden cambiar con facilidad, y al hecho de que la navegación requiere una definición de enlaces flexible, que pueda ser cambiada de forma dinámica. Por lo tanto, con este esquema se puede mejorar la mantenibilidad y la flexibilidad.

Así, proponemos el reflejo de la entidad `Enlace` que aparece en el metamodelo de UWE (Figura 5) directamente en implementación. Para ello, necesitamos

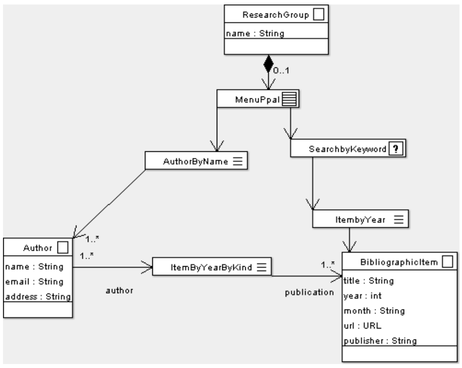

Figura 4. Modelo navegacional del ejemplo de gestión de publicaciones

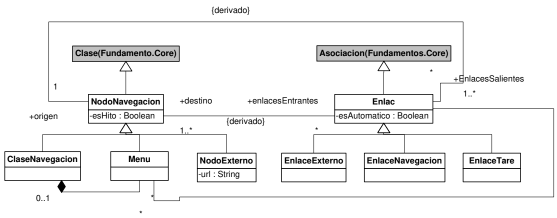

Figura 5. Paquetes del metamodelo de UWE

definir una clase `Enlace` en el espacio de los meta-objetos. De esta forma, la información sobre la relación entre entre nodos se encapsula y se separa de los propios nodos.

Basándonos en el razonamiento presentado en SADES, los filtros de composición son los candidatos ideales para situaciones en las que hay que interceptar mensajes. Por lo tanto, tendremos que adaptar el mecanismo de intercepción de mensajes para que cualquier mensaje que manipule enlaces entre nodos de navegación sea recibido por el objeto `Enlace` y no por las entidades conectadas mediante estos enlaces.

5. Conclusiones y trabajo futuro

En este artículo presenta un acercamiento en una fase muy inicial a la implementación de un sistema generador de aplicaciones web, que tiene como requisitos la separación de conceptos durante todo su desarrollo. Como han hecho algunas de las metodologías de ingeniería web, se propone extender una idea aparecida en el campo de las bases de datos a los sistemas web. Esta idea surge debido a la observación del mismo tipo de elementos en las bases de datos que en los sistemas web.

Utilizando el enfoque de relaciones dinámicas para el desarrollo de sistemas web se pueden obtener sistemas más flexibles y fáciles de mantener.

Aunque esta la idea presentada en el artículo está en un estadio de desarrollo muy temprano, se pretende seguir profundizando en esta línea, así como integrar las conclusiones obtenidas en esta línea, en las construcción de un framework conducido por modelos, y basado en la filosofía MDA.

Referencias

1. H. Baumeister, N. Koch, and L. Mandel. Towards a uml extension for hypermedia design. In *UML99 The Unified Modelling Language - Beyond the Standard*. Springer-Verlag Lecture Notes in Computer Science, October 1999.
2. L. Bergmans and M. Aksit. *Aspect-Oriented Software Development*, chapter 4, pages 63–95. Addison-Wesley, 2005.
3. O. de Troyer and C. Leuner. Wsdm: A user centered design method for web sites. In *Proceedings of the 7th International World Wide Web Conference*. Elsevier, 1998.
4. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns, Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.
5. J. Gómez, C. Cachero, and O. Pastor. Extending a conceptual modelling approach to web application design. In *Proceedings of the Conference on Advanced Information Systems Engineering*, pages 79–93, 2000.
6. G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C.V. Lopes, J. M. Loingtier, and J. Irwin. Aspect-Oriented Programming. In *proceedings of the European Conference on Object-Oriented Programming*. Springer-Verlag, June 1997.
7. A. Knapp, N. Koch, F. Moser, and G. Zhang. Argouwe: A case tool for web applications. In *Proceedings of the First Int. Workshop on Engineering Methods to Support Information Systems Evolution (EMSISE´03)*, Sept 2003.
8. N. Koch and A. Kraus. Towards a common metamodel for the development of web application. In *Proceedings of the Third International Conference on Web Engineering (ICWE 2003)*. Springer-Verlag Lecture Notes in Computer Science, July 2003.
9. A. Kraus and N. Koch. A metamodel for uwe. Technical Report 0301, Institut fur Informatik, Ludwig-Maximilians-Universitat Munchen, January 2003.
10. Mira Mezini and Karl Lieberherr. Adaptive plug-and-play components for evolutionary software development. In C. Chambers, editor, *OOPSLA*, pages 97–116. ACM, October 1998.
11. A. Rashid and P. Sawyer. Dynamic relationships in object oriented databases: A uniform approach. In *Proceedings of the Database and Expert Systems Applications (DEXA 99)*, number 1677, pages 26–35. Springer-Verlag Lecture Notes in Computer Science, 1999.
12. Awais Rashid. A hybrid approach to separation of concerns: The story of SADES. In A. Yonezawa and S. Matsuoka, editors, *Metalevel Architectures and Separation of Crosscutting Concerns 3rd Int'l Conf. (Reflection 2001), LNCS 2192*, pages 231–249. Springer-Verlag, September 2001.
13. A. M. Reina, J. Torres, M.J. Escalona, and J. A. Ortega. Revisiting requirements in web modelling languages. In *Proceedings of the IADIS WWW/Internet 2003 Conference*, 2003. Aceptado para publicación.
14. D. Schwabe, G. Rossi, and S. Barbosa. Systematic Hypermedia Design with OOHDM. In *ACM International Conference on Hypertext' 96*, 1996.