

Migración Dirigida por Modelos de Sistemas Web Heredados a Rich Internet Applications*

Roberto Rodríguez-Echeverría, José María Conejero, Pedro J. Clemente, and
Juan C. Preciado

Universidad de Extremadura,
Quercus Software Engineering Group
{rre, chemacm, pjcleme, jcpreciado}@unex.es
<http://quercusseg.unex.es>

Abstract. Actualmente existe una tendencia de migración de aplicaciones Web heredadas (AWHs) a RIAs, fundamentada en las mejoras que introducen las tecnologías RIA tanto en la interacción con el usuario como en la optimización de la conexión con el servidor. Sin embargo, estos procesos de reingeniería se han realizado tradicionalmente de manera ad-hoc, resultando en proyectos con unos costes muy altos y baja reusabilidad. Esta situación está parcialmente motivada por el hecho de que la mayoría de las AWHs se construyeron antes de que las principales propuestas de Desarrollo Software Dirigidas por Modelos (DSDM) alcanzasen una madurez suficiente. La OMG Architecture-Driven Modernization (ADM) aboga por la aplicación de principios DSDM con el objetivo de formalizar y estandarizar estos procesos de reingeniería. Este trabajo presenta un proceso sistemático de migración de AWHs a RIAs basado en los principios de ADM.

Keywords: Ingeniería Web, Reingeniería Software, RIA.

1 Introducción

Las Rich Internet Applications (RIA) han emergido como la plataforma más prometedora para el desarrollo de aplicaciones Web 2.0, gracias a su potente soporte multimedia, riqueza de presentación e interacción, trabajo colaborativo y flexibilidad en la comunicación, entre otras características [7].

Actualmente existe una clara tendencia actual en la industria del software a migrar de AWHs hacia RIAs. La gran mayoría de esas AWHs se desarrollaron antes que las principales propuestas de Desarrollo Web Dirigido por Modelos [9] estuviesen lo suficientemente maduras para su adopción por la industria. Como principales inconvenientes de esta situación, nos encontramos con AWHs que carecen de una documentación completa y actualizada, y que han seguido procesos de mantenimiento de baja calidad. En este escenario tan complejo,

* Trabajo financiado por el MEC (TIN2008-02985), fondos FEDER y la Junta de Extremadura

la industria demanda la formalización y estandarización de estos procesos de reingeniería para reducir sus costes y riesgos. Precisamente, ADM propone la aplicación de principios, técnicas y herramientas de Desarrollo Software Dirigido por Modelos con este fin.

El objetivo fundamental de este trabajo es definir un marco de trabajo flexible para la migración sistemática de AWHs en RIAs haciendo un uso exhaustivo de métodos, técnicas y herramientas de DSDM. En concreto, en este artículo, se presentan los pasos fundamentales del proceso, prestando especial atención a las primeras fases de extracción de información y refinado del sistema conceptual.

El resto del artículo se organiza como se describe a continuación. La sección 2 presenta una visión general de la propuesta. La sección 3 ilustra la propuesta con un sencillo ejemplo. La sección 4 comenta algunos trabajos relacionados. Finalmente, la sección 5 enumera las conclusiones e introduce el trabajo futuro.

2 Propuesta

Como muestra la figura 1, nuestro proceso de migración tiene como objetivos principales: (1) generar un cliente RIA; y (2) en el lado del servidor, generar una capa orientada a servicios que permita la conexión del nuevo cliente RIA con la lógica de negocio de la AWH.

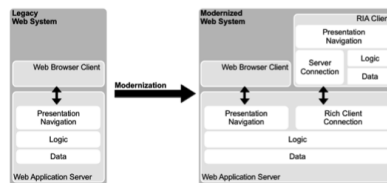


Fig. 1. Arquitectura del sistema final

El cliente RIA podrá estar formado por una interfaz de usuario rica (más interactiva), por los datos almacenados en el lado cliente, por la lógica de negocio procesada en el cliente, así como por la infraestructura lógica para la comunicación y sincronización con el servidor. La parte servidora no se debe ver modificada, de manera que el sistema pueda seguir ofreciendo toda la lógica de negocio de la Web. Con este objetivo, se creará una capa de conexión entre el cliente RIA y la lógica de negocio original. Esta capa será la encargada de gestionar la distribución de datos y lógica en el cliente e integrar un modelo de comunicación asíncrono entre el nuevo cliente y el servidor.

Como muestra la figura 2, planteamos un proceso de reingeniería compuesto básicamente de dos etapas: (1) obtener una representación abstracta del sistema mediante técnicas de ingeniería inversa; y (2) generar el sistema software final a partir de la representación obtenida. Partiendo de una aplicación Web heredada,

se obtiene una representación conceptual de los elementos relevantes de la misma para asistir en su migración a una RIA.

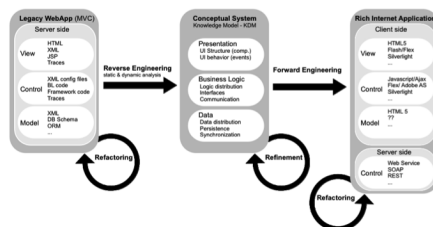


Fig. 2. Visión general de la propuesta de migración

La primera fase de nuestro proceso de migración intenta reducir la complejidad del proceso de migración pasando de la heterogeneidad de las tecnologías de implementación a la homogeneidad de los modelos. Con este propósito, se utilizan soluciones disponibles para la transformación de código a modelo (análisis estático), tales como los discoveres y los metamodelos de MoDisCo¹ para Java, JSP y XML (Specific Abstract Syntax Tree Metamodels, SASTM²). Como productos finales de esta fase se obtienen modelos dependientes de la tecnología que representan la AWH completa. Por lo tanto, a partir de este momento, se puede trabajar exclusivamente con modelos.

Otra actividad fundamental dentro de la fase de extracción de información consiste en la aplicación de análisis dinámicos a la AWH. Consideramos que el análisis de las trazas de ejecución puede proporcionarnos información suficiente de la interacción del usuario con la AWH, de manera que se puedan apoyar las decisiones relativas a la generación del cliente RIA.

Una vez obtenida una primera representación del sistema con modelos dependientes de la tecnología, se generará un modelo de conocimiento independiente de la tecnología (KDM³) mediante la aplicación de transformaciones M2M, siguiendo la propuesta de [1,2,6]. A partir de esta representación conceptual del sistema heredado, se puede acometer una secuencia de fases de refinado para dotarlo de información suficiente que asista en el proceso de generación del sistema final. Como actividad fundamental dentro de esta fase, destaca la localización de expresiones de patrones RIA característicos.

Finalmente, mediante técnicas y herramientas de ingeniería directa se pretende generar el sistema objetivo (cliente RIA y capa de conexión) haciendo uso de propuestas y herramientas MDWE con extensiones RIA suficientemente validadas, como ejemplifica [3] con WebRatio y RUX-Tool.

¹ <http://www.eclipse.org/MoDisco/>

² <http://www.omg.org/spec/ASTM/>

³ Knowledge Discovery Metamodel - <http://www.omg.org/technology/kdm/index.htm>

3 Ejemplo

Como ejemplo ilustrativo de AWH, hemos escogido JAVA Pet Store⁴ Demo (Petstore). En concreto, en este caso vamos a centrarnos en dos de las páginas que implementan su catálogo, página product y página item, que se muestran en la figura 3. Como se puede observar, existe una clara relación Maestro/Detalle entre los datos presentados por la sección principal de ambas páginas (el listado de los productos de tipo Perro y los detalles del producto Chihuahua macho adulto).



Fig. 3. Relación Maestro/Detalle entre páginas *product* e *item*.

A partir de la solución multipágina presentada por Petstore, pasando por las fases de extracción y representación del conocimiento de nuestra propuesta, se obtiene su representación en KDM, como muestra la figura 4 (sólo paquetes Code y UI). Con objetivo de refinar el modelo conceptual del sistema, se realiza una actividad de búsqueda del patrón Maestro/Detalle multipágina. Cada vez que se encuentra este patrón se marca convenientemente el modelo para su identificación.

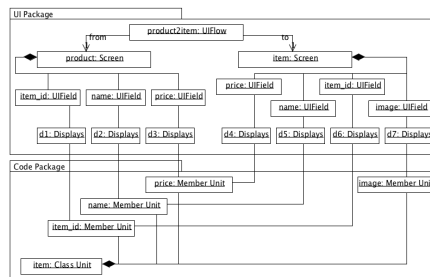


Fig. 4. Maestro/Detalle multipágina en KDM (simplificación).

Posteriormente, el ingeniero encargado de la migración podrá decidir qué solución RIA aplica a esta situación (patrones RIA) en cada uno de los casos identificados. En el caso de relaciones Maestro/Detalle, los patrones RIA más

⁴ Version1.3.2:[Aug04,2003]http://java.sun.com/blueprints/code/jps132/docs/index.html

comunes son, como ilustra la figura 5: Master/Detail Screen Pattern [11] y Quicklook pattern. La aplicación de un patrón RIA supondrá la reestructuración del modelo conceptual del sistema heredado.

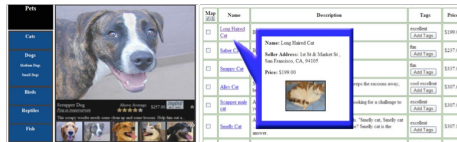


Fig. 5. Patrones RIA para relación Maestro/Detalle

4 Trabajos relacionados

Durante la última década, como se señala en [5], se han desarrollado importantes trabajos en el área de la ingeniería inversa de aplicaciones Web. VAQUISTA [12] aplica técnicas de ingeniería inversa para facilitar la migración del interfaz de usuario de una aplicación Web a diferentes plataformas. [4] aplica técnicas de ingeniería inversa para migrar el interfaz multipágina de una aplicación Web a un interfaz de usuario Web 2.0 (una sola página). Todos estos trabajos presentan una estrecha relación con la fase de ingeniería inversa de nuestro proceso de migración.

En [8,10] se presentan propuestas para la incorporación sistemática de funcionalidades RIA a aplicaciones Web heredadas. La diferencia fundamental con este trabajo consiste en que esas propuestas parten exclusivamente de aplicaciones Web desarrolladas mediante propuestas DSDM.

Por otra parte, como ejemplo de la aplicación de principios y técnicas DSDM en procesos de mantenimiento, cabe destacar la propuesta de [1]. MoDisco es una solución de modernización de software genérica, extensible y open source, que hace un uso intensivo de principios y técnicas de DSDM. Nuestra propuesta pretende especializar el marco de trabajo definido por MoDisco para el escenario concreto de migración de sistemas Web heredados a RIAs.

5 Conclusiones y trabajo futuro

Este trabajo presenta una propuesta para la definición de un proceso sistemático de migración de aplicaciones Web heredadas (AWHs) a RIAs, utilizando principios, técnicas y herramientas del DSDM. Concretamente, el proceso de migración presentado tiene como objetivo principal generar un cliente RIA a partir de las capas de presentación y navegación de la AWH, fundamentalmente, construyendo además una capa de conexión orientada a servicios en la parte servidora.

Dada la extensión y complejidad de cualquier proceso de migración, existe un gran número de líneas de trabajo que acometer. Entre ellas, estamos especialmente interesados en cuatro: (1) aumentar el catálogo de patrones RIA para el proceso de refinado del sistema conceptual; (2) diseñar análisis dinámicos que permitan obtener información suficiente para asistir en la distribución de lógica de negocio y datos entre cliente y servidor; (3) extender la aplicación de otras especificaciones de la ADM en el proceso; y (4) proporcionar herramientas de soporte para este proceso.

References

1. Hugo Bruneliere, Jordi Cabot, and Frédéric Jouault. MoDisco : A Generic And Extensible Framework For Model Driven Reverse Engineering. In *IEEE/ACM International Conference on Automated Software Engineering*, pages 1–2, 2010.
2. Javier Luis Cánovas Izquierdo and Jesús García Molina. An Architecture-Driven Modernization Tool for Calculating Metrics. *IEEE Software*, 27(4):37–43, 2010.
3. Marino Linaje, Juan Carlos Preciado, Rober Morales-Chaparro, Roberto Rodríguez-Echeverría, and Fernando Sánchez-Figueroa. Automatic Generation of RIAs Using RUX-Tool and Webratio. 9th International Conference on Web Engineering, 501–504. San Sebastian, 2009.
4. Ali Mesbah and Arie van Deursen. Migrating Multi-page Web Applications to Single-page AJAX Interfaces. *11th European Conference on Software Maintenance and Reengineering (CSMR'07)*, pages 181–190, March 2007.
5. Reshma Patel, Frans Coenen, Russell Martin, and Lawson Archer. Reverse Engineering of Web Applications: A Technical Review. Technical Report July 2007, University of Liverpool Department of Computer Science, Liverpool, 2007.
6. Ricardo Pérez-Castillo, Ignacio García-Rodríguez De Guzmán, and Mario Piattini. Business Process Archeology using MARBLE. *Information and Software Technology*, 2011.
7. Juan Carlos Preciado, Marino Linaje, Fernando Sanchez, and Sara Comai. Necessity of methodologies to model Rich Internet Applications. *Seventh IEEE International Symposium on Web Site Evolution*, 2005.
8. Roberto Rodríguez-Echeverría, José María Conejero, Marino Linaje, Juan Carlos Preciado, and Fernando Sánchez-Figueroa. Re-engineering legacy Web applications into Rich Internet Applications. In *10th International Conference on Web Engineering*, 2010.
9. Gustavo Rossi, Oscar Pastor, Daniel Schwabe, and Luis Olsina. Web Engineering: Modelling and Implementing Web Applications (Human-Computer Interaction Series). October 2007.
10. Gustavo Rossi, Matias Urbieto, Jeronimo Ginzburg, Damiano Distante, and Alejandra Garrido. Refactoring to Rich Internet Applications. A Model-Driven Approach. *2008 Eighth International Conference on Web Engineering*, pages 1–12, July 2008.
11. Bill Scott and Theresa Neil. *Designing Web Interfaces: Principles and Patterns for Rich Interactions*. O'Reilly Media, 2009.
12. J. Vanderdonckt, L. Bouillon, and N. Souchon. Flexible reverse engineering of web pages with VAQUISTA. *Proceedings Eighth Working Conference on Reverse Engineering*, pages 241–248, 2001.