

Modelado de la variabilidad en arquitecturas multicapa ^{*}

José García-Alonso, Joaquín Guillén, Javier Berrocal, and Juan Manuel Murillo

Escuela Politécnica, Universidad de Extremadura,
Avd. de la Universidad s/n, 10071 Cáceres, Spain
jgaralo@unex.es, jguillen@gloin.es, jberrocal@unex.es, juanmamu@unex.es

Abstract. La tendencia de la industria a la hora de abordar el desarrollo de aplicaciones multicapa se ha dirigido a la creación y utilización de complejos frameworks de desarrollo. La utilización de estos frameworks requiere una gran especialización por parte de los arquitectos software que deben conocer en profundidad las capacidades de un gran número de ellos. Esto hace que el éxito de los desarrollos dependa en gran medida de la experiencia, la pericia y los posibles errores que pueda cometer el arquitecto. En este artículo se presenta un modelo de características que captura la variabilidad arquitectónica y tecnológica de este tipo de aplicaciones y que es utilizado como eje central de una línea de productos. Esta línea de productos facilita al arquitecto la labor de convertir los requisitos iniciales de una aplicación en un diseño específico de una arquitectura multicapa implementada con frameworks, haciendo este proceso más adaptable a posibles evoluciones arquitectónicas y tecnológicas.

Keywords: Arquitecturas multicapa, variabilidad, desarrollo dirigido por modelos, frameworks de desarrollo

1 Introducción

Las arquitecturas multicapa y los patrones de diseño proporcionan la infraestructura necesaria para construir aplicaciones fiables y fáciles de evolucionar. Se ha invertido mucho esfuerzo en diseñar métodos que simplifiquen la aplicación de dichos patrones y arquitecturas. El resultado ha sido la aparición de un gran número de frameworks de desarrollo [1]. Actualmente, el uso de estos frameworks está ampliamente extendido en la industria del software, prueba de ello es el elevado número de los mismos disponibles [2], las ofertas de trabajo que requieren experiencia en ellos o el número de versiones que se liberan al año [3].

El uso de estos sistemas no solo requiere de una gran especialización por parte de los desarrolladores sino que la figura del arquitecto software adquiere especial relevancia. Sobre él se deposita una gran responsabilidad, de forma que el éxito de los desarrollos depende en gran medida de su experiencia, su pericia y de los posibles errores que pudiese cometer [4]. Especialmente al adaptar los

^{*} Este trabajo ha sido financiado por el proyecto TIN2008-02985

requisitos de la aplicación a desarrollar a los patrones de diseños y frameworks que se usarán en la implementación.

Numerosos trabajos, especialmente en el área de la ingeniería web, abordan la complejidad en el desarrollo de aplicaciones multicapa basadas en frameworks. Para ello, se restringen las posibilidades arquitectónicas y tecnológicas de los desarrollos simplificando así la labor del arquitecto.

En este artículo se presenta un modelo de características que captura la variabilidad arquitectónica y tecnológica de las aplicaciones multicapa. Este modelo se utiliza como eje central de una línea de productos en la que, partiendo de los requisitos iniciales de una aplicación y mediante una serie de transformaciones, se obtiene un diseño específico para la arquitectura y tecnología elegidas. De este modo, la combinación de técnicas propias de las líneas de producto y el desarrollo dirigido por modelos permitirán guiar al arquitecto en el proceso de obtener un diseño específico para la arquitectura elegida, disminuyendo los posibles errores que pudiera cometer y simplificando la adaptación del mismo a evoluciones arquitectónicas o tecnológicas.

En la sección dos se detallan los antecedentes de este trabajo. En la sección tres se describe el modelo de características. En la sección cuatro se esboza el proceso propuesto para la obtención de modelos específicos partiendo de los requisitos iniciales. En la sección cinco se detallan los trabajos relacionados. Finalmente, en la sexta sección aparecen las conclusiones y trabajos futuros.

2 Antecedentes

Numerosos trabajos en el campo de ingeniería web tratan con el desarrollo de sistemas basados en frameworks. Trabajos como WebML [5], Rux [6] y otros permiten desarrollar aplicaciones basadas en frameworks de desarrollo. El uso de las técnicas de desarrollo dirigido por modelos en estos trabajos simplifica en gran medida el trabajo de los arquitectos software. Esta simplificación se consigue, en parte, eliminando la variabilidad arquitectónica y tecnológica. La arquitectura de los desarrollos realizados utilizando estas técnicas está implícita en los modelos utilizados. Las tecnologías a utilizar se limitan a aquellas soportadas por los motores de transformación de modelo a código proporcionados. La restricción de la variabilidad supone una serie de limitaciones a la hora de utilizar dichas técnicas, permitiéndose solo los desarrollos que se adaptan a las arquitecturas o tecnologías implícitas.

En este artículo se presenta un modelo de características que permite incorporar la variabilidad arquitectónica y tecnológica a los desarrollos de este tipo. Para representar dicho modelo de variabilidad se utiliza Cardinality-Based Feature Modeling (CBFM) [7]. Se ha elegido esta técnica de modelado de variabilidad por ser una extensión de FODA [8], por disponer de soporte de herramientas modelado [9] y por haber demostrado su utilidad a la hora de trabajar con frameworks de desarrollo [10].

3 Modelo de características

En esta sección se presenta el modelo de características que se propone para capturar la variabilidad arquitectónica y tecnológica de las aplicaciones multicapa. Por motivos de espacio no se presenta una imagen completa de dicho modelo ¹

Este modelo se organiza en distintos niveles jerárquicos. Cada uno de estos niveles representa un nivel de abstracción en una arquitectura multicapa. El primer nivel comprende las capas que pueden incluirse en la aplicación. En la figura 1 se observan las capas soportadas actualmente. Tanto el modelo como el proceso de transformaciones que se presenta en la próxima sección pueden ser extendidos para incluir nuevas capas o tecnologías que sean necesarias en las aplicaciones a desarrollar.

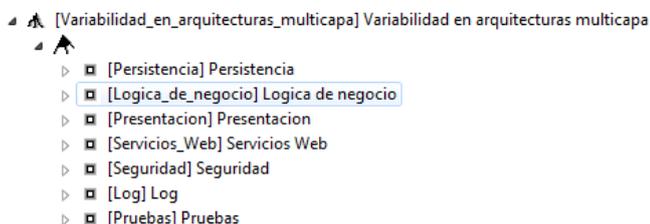


Fig. 1. Modelo de variabilidad para arquitecturas multicapa. Nivel de capa

Cada una de las capas de este modelo contiene, en el segundo nivel jerárquico, los patrones de diseño o técnicas más utilizados para la implementación de dicha capa. En el tercer nivel del modelo se detallan las opciones tecnológicas disponibles para la implementación de un determinado patrón de diseño o técnica. En la figura 2 se observa la parte del modelo correspondiente a la capa de presentación. Para la implementación de esta capa se permite utilizar tres patrones de diseño, el clásico MVC y dos patrones propios de la web 2.0, Web Remoting y Page Rearrangement. Cada uno de estos patrones puede implementarse utilizando distintas tecnologías.

En algunas ocasiones el modelo contiene un cuarto nivel jerárquico. En este nivel se presentan las distintas posibilidades de utilización de una tecnología para implementar un patrón de diseño.

Cada uno de los niveles de este modelo puede configurarse en distintas fases [11] del desarrollo de aplicaciones multicapa, eligiendo para cada nivel jerárquico y desarrollo concreto las características más adecuadas. Sin embargo, el modelo en sí mismo solo contiene una taxonomía de patrones de diseño y tecnologías que pueden utilizarse en el desarrollo de estas aplicaciones. En la siguiente sección se

¹ Puede consultarse el modelo completo en: <http://www.gloin.es/jisbd2011/VariabilidadArquitecturasMulticapa.fmp>. Para poder visualizar y manipular este modelo se necesita el plugin de Eclipse que puede encontrarse en: <http://gsd.uwaterloo.ca/node/18>

El primer paso de este proceso consiste en anotar los requisitos de la aplicación con los atributos de calidad (QA) que se quieren conseguir en la implementación de la misma. Se puede encontrar más información sobre este punto en un trabajo de los mismos autores [12].

Utilizando los requisitos anotados de la aplicación se configura el primer nivel jerárquico del modelo de variabilidad y se presenta al arquitecto el conjunto de capas que se proponen para desarrollar la aplicación. Una vez que el arquitecto valida las capas propuestas, se utilizan los requisitos iniciales de la aplicación y el modelo de variabilidad con la configuración del primer nivel jerárquico para generar una transformación de modelos. Esta transformación se aplica a los requisitos iniciales para obtener una versión de los requisitos adaptados a las capas que van a componer la aplicación.

Un estudio de los requisitos en mayor detalle, junto a la ayuda del arquitecto, permiten configurar el segundo nivel jerárquico del modelo de variabilidad. Con esta nueva configuración del modelo de características se obtiene una nueva transformación que proporciona un modelo de los requisitos adaptados a los patrones de diseño y técnicas que se van a utilizar para la implementación.

A continuación se presentan al arquitecto o se eligen automáticamente, según algún criterio definido, las opciones tecnológicas disponibles para implementar la aplicación. Por último se decide cómo utilizar las tecnologías elegidas de entre las opciones disponibles en el modelo de características. En cada uno de estos pasos se aplican sucesivas transformaciones a los modelos de la aplicación para refinarlos hasta llegar a disponer de modelos específicos de la arquitectura y tecnologías elegidos.

Estos modelos finales pueden entregarse directamente a los desarrolladores o utilizarse como entrada de un sistema de generación de código.

5 Trabajos relacionados

Numerosos trabajos en el área del desarrollo de software dirigido por modelos tratan con el desarrollo de sistemas con arquitecturas complejas y cuyo código utiliza frameworks de desarrollo. Especialmente en el ámbito de la ingeniería web con trabajos como WebML [5] o RUX [6].

Estos trabajos reducen la complejidad de los desarrollos limitando las decisiones arquitectónicas y tecnológicas disponibles, con una excepción, los trabajos de Meliá y Gómez [13], [14]. En estos trabajos los autores proponen una extensión a los métodos anteriores. El propósito de esta extensión es flexibilizar estos métodos incorporando a los mismos la posibilidad de definir la arquitectura que se quiere utilizar. Para ello se centran en aplicaciones web y en RIA respectivamente, mientras que el trabajo presentado aquí pretende abarcar un mayor rango de aplicaciones. Adicionalmente, al contrario que en este trabajo, los trabajos aquí mencionados no permiten a los desarrolladores decidir acerca de las tecnologías que van a ser utilizadas para la implementación de las aplicaciones.

6 Conclusiones y trabajos futuros

En este artículo se ha presentado el esbozo una técnica para convertir los requisitos iniciales de una aplicación en diseños específicos de una arquitectura multi-capa concreta que utiliza tecnología y patrones de diseño concretos. Para ello se utiliza un modelo de características que captura la variabilidad arquitectónica y tecnológica de este tipo de aplicaciones y técnicas de desarrollo dirigido por modelos y líneas de producto para ir refinando los distintos modelos de la aplicación a desarrollar.

Este trabajo se encuentra en un estado inicial. Se debe refinar el modelo de características descrito para dar soporte a un mayor número de capas, patrones de diseño y tecnologías. Se debe proporcionar soporte de herramientas para el conjunto de transformaciones descrito en la cuarta sección. Finalmente, se debe proporcionar mecanismos específicos para la generación del código de las aplicaciones a partir de los modelos obtenidos mediante las distintas transformaciones.

References

1. Johnson, R. J2EE development frameworks. *Computer*, vol.38, no.1, 2005
2. [Shan06] Shan, T., Hua, W. Taxonomy of Java Web Application Frameworks. *e-Business Engineering*, 2006. ICEBE '06. IEEE International Conference on , vol., no., pp.378-385, Oct. 2006
3. Raible, M. Comparing Java Web Frameworks. Apache convention NA, 2007. Web site - <http://static.raibledesigns.com/repository/presentations/ComparingJavaWebFrameworks-ApacheConUS2007.pdf>
4. Dalgarno, M. When Good Architecture Goes Bad. *Methods & Tools - Spring 2009*
5. Brambilla, M., Comai, S., Fraternali, P., Matera, M. Web Engineering: Modelling and Implementing Web Applications chap. Designing Web Applications with Webml and Webratio. *Human Computer Interaction Series*. 2008, 221-261
6. Linaje, M., Preciado, J. C., Sánchez-Figueroa, F. Engineering Rich Internet Application User Interfaces over Legacy Web Models. *IEEE Internet Computing* (2007)
7. K. Czarnecki, S. Helsen, U. Eisenecker, Formalizing cardinality-based feature models and their specialization, *Software Process Improvement and Practice* (2005)
8. K. Kang, S. Cohen, J. Hess, W. Novak, S. Peterson, Feature Oriented Domain Analysis (FODA) Feasibility Study, Technical Report CMU/SEI-90-TR-021, 1990.
9. M. Antkiewicz, K. Czarnecki, FeaturePlugin: Feature Modeling Plug-In for Eclipse, in: *Proceedings of the 2004 OOPSLA workshop on eclipse technology eXchange*.
10. Antkiewicz, M., Czarnecki, K., Stephan, M. Engineering of Framework-Specific Modeling Languages. *IEEE Trans. Software Eng.* 35(6): 795-824 (2009)
11. Czarnecki, K., Helsen, S., Eisenecker, U. W. Staged configuration through specialization and multilevel configuration of feature models. *Software Process: Improvement and Practice* 10(2): 143-169 (2005)
12. Berrocal, J., García-Alonso, J., Murillo, J.M. Facilitating the selection of architectural patterns by means of a marked requirements model. In *Proc. 4th European Conference on Software Architecture*. 2010. Pp. 384-391
13. Meliá, S., Gómez, J. The WebSA Approach: Applying Model Driven Engineering to Web Applications. *J. Web Eng.* 5(2): 121-149 (2006)
14. Meliá, S., Gómez, J., Pérez, S., Díaz, O. Architectural and Technological Variability in Rich Internet Applications. *IEEE Internet Computing* 14(3): 24-32 (2010)