

# Un enfoque Multi-modelo para la Introducción de Atributos de Calidad en el Desarrollo de Líneas de Producto Software

Javier Gonzalez-Huerta, Silvia Abrahão, Emilio Insfran

Grupo de investigación ISSI  
Depto. de Sistemas Informáticos y Computación.  
Universitat Politècnica de València  
Camino de Vera s/n, 46022, Valencia  
{jagonzalez, sabrahao, einsfran}@dsic.upv.es

**Resumen.** La aplicación del enfoque de Líneas de Producto Software está aplicándose, a la par que extendiéndose, de manera notable en la industria. Su adopción incrementa la reutilización de diferentes activos durante el proceso de desarrollo de software. En muchos ámbitos la aplicación de este enfoque va de la mano de procesos de desarrollo dirigidos por modelos. Este artículo presenta una propuesta para integrar la calidad como una vista más del sistema, permitiendo además establecer relaciones entre las diferentes vistas del sistema (variabilidad, funcionalidad y calidad). La factibilidad de la propuesta se ilustra a través de un ejemplo en el contexto de la Ingeniería de Sistemas basada en Modelos en el dominio de los Sistemas Críticos de Seguridad.

**Palabras clave:** Líneas de Producto Software, Desarrollo de Software Dirigido por Modelos, Ingeniería de Sistemas, Aseguramiento de la Calidad.

## 1. Introducción

La producción de software de calidad, en el tiempo adecuado y con unos costes razonables es un problema recurrente de la Ingeniería del Software abordado ya desde distintas aproximaciones. Una solución al problema aplicada a nivel industrial consiste en emplear Líneas de Producto Software (LPS). Una LPS es un conjunto de sistemas software que comparten un conjunto de características comunes (*features*) que satisfacen las necesidades específicas de un dominio o segmento de mercado particular, y que se desarrollan a partir de una arquitectura base y un conjunto de activos software comunes (*core assets*), de un modo preestablecido y sistemático mediante un plan de producción [2]. El desarrollo de LPS consta de dos procesos básicos: la *Ingeniería del Dominio*, donde se establece las partes comunes y las variables y se construye un conjunto de activos, y la *Ingeniería de la Aplicación*, donde los activos son reutilizados para derivar productos específicos.

Pero el éxito de un producto software no radica únicamente en el cumplimiento de los requerimientos funcionales, sino también en los atributos de calidad que los

productos deben tener. Los trabajos actuales sobre LPS se centran normalmente en una única vista del sistema (la vista de variabilidad comúnmente representada mediante Modelos de Características) que por sí sólo no define la extensión de una LPS. La gestión de la variabilidad en las LPS, implica por un lado el manejo de las características del dominio plasmadas en un modelo de características, y por otro lado que dicha variabilidad debe ser soportada en activos software. Para asegurar la calidad del producto software, esta debería ser tenida en cuenta en el proceso de desarrollo desde fases tempranas. Esto nos lleva a que consideraremos (al menos) tres vistas de los sistemas software bajo la perspectiva de líneas de producto software: i) la Vista de la Variabilidad del Sistema (VVS); ii) la Vista Funcional del Sistema (VFS); y, iii) la Vista de la Calidad del Sistema (VCS).

En trabajos previos se presentó una arquitectura [5], un ejemplo de automatización [6] y unas guías de diseño [7] para la introducción de atributos de calidad como factor de decisión en procesos de transformación de modelos. En este trabajo, pretendemos introducir esta propuesta en una aproximación de Líneas de producto.

En un contexto de líneas de producto software, una característica es un conjunto de requerimientos relacionados que constituyen una unidad lógica observable para el uso del producto. Un bloque funcional puede ser visto como el comportamiento del sistema que satisface las necesidades de una o más características. Esta puede satisfacer enteramente la característica mediante la combinación de otras funciones. Del mismo modo, una función puede estar presente en varias características. Es por ello que la relación entre características y funcionalidades dista mucho de ser una relación uno a uno.

Si atendemos a la perspectiva de calidad, las características seleccionadas en una configuración específica van a tener un impacto directo sobre los atributos de calidad del producto. Es más, el modo en que las características son mapeadas en bloques funcionales va a tener influencia sobre los atributos de calidad del producto final.

Actualmente no hay propuestas que permitan expresar relaciones entre las distintas características y el impacto de éstas sobre los atributos de calidad, así como relaciones entre características y funcionalidades y de estas últimas con los atributos de calidad.

Este artículo presenta una propuesta para integrar la vista calidad como una vista más del sistema, permitiendo además establecer relaciones entre el modelo de características (que representa la vista de variabilidad) y el modelo de calidad, y las relaciones existentes entre la vista de variabilidad y la vista funcional del sistema siguiendo una estrategia multi-modelo. Un multi-modelo es una colección de modelos que dan soporte a diferentes vistas de un sistema, caracterizados por la existencia de relaciones entre los elementos de sus correspondientes metamodelos [1]. El multi-modelo constará de varios modelos o vistas del sistema (de funcionalidad, de características, de calidad...) con relaciones entre ellas (por ej. características con calidades diferentes) que nos lleva a la parametrización del proceso de producción de software mediante dicho multi-modelo.

El trabajo se centra en las LPS para el desarrollo de sistemas software críticos (*safety-critical embedded systems*) en el contexto de la Ingeniería de Sistemas. El término ingeniería de sistemas hace referencia a la necesidad de identificar y manejar las propiedades de un sistema como un todo. En proyectos de gran complejidad puede ir más allá de la suma de sus partes. Los métodos y técnicas de modelado aplicadas a la ingeniería de sistemas ayudan a una mejor comprensión de los mismos. La

ingeniería de sistemas frecuentemente hace frente a problemas en los que está presente la componente software y la hardware, sin embargo, nuestros esfuerzos se centran en resolver problemas desde el punto de vista del software, empleando el paradigma de desarrollo de software dirigido por modelos (DSDM), ya que todos los principios aplicados en este trabajo pueden ser extendidos al contexto de Ingeniería de Sistemas Basada en Modelos (*model-based systems engineering* - MBSE).

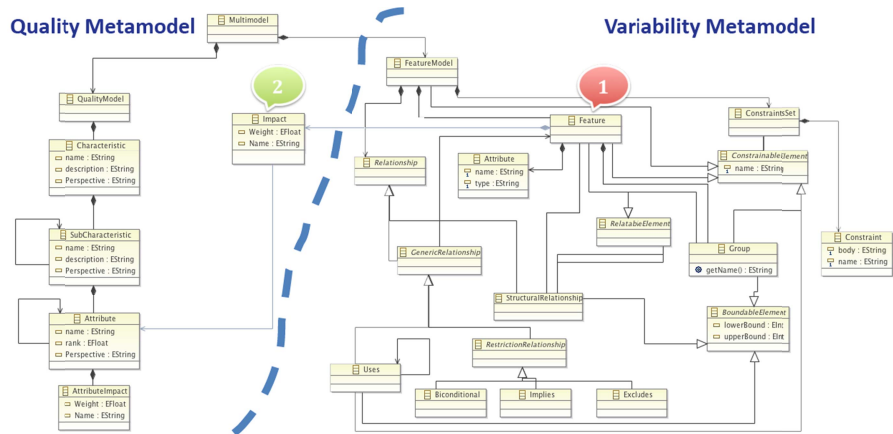
Algunos trabajos que pueden encontrarse en la literatura se enfrentan al problema en el dominio de los sistemas críticos, pero la mayoría de ellos lo hacen de manera parcial o como en [8], [11], donde se presenta un enfoque DSDM basado en LPS pero sin tener en cuenta aspectos no funcionales.

La organización del artículo es la siguiente: la sección 2 presenta la definición de un multi-modelo para la introducción de atributos de calidad en el desarrollo de líneas de producto software. La sección 3 presenta un ejemplo que ilustra la aplicabilidad de la propuesta. Finalmente, la sección 4 presenta las conclusiones y trabajos futuros.

## **2. Multi-modelo para la Introducción de Atributos de Calidad en el Desarrollo de Líneas de Producto Software**

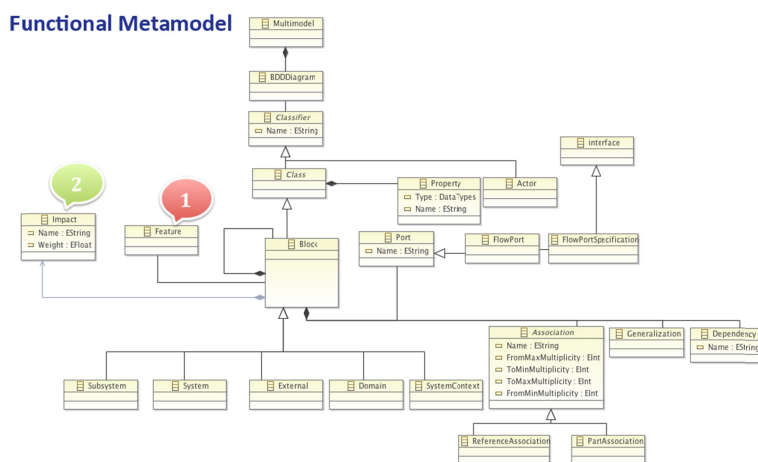
Nuestra propuesta se centra en la definición de un multi-modelo para poder representar las características de calidad y las relaciones que éstas tienen con las distintas vistas del sistema. Estas relaciones se tendrán en cuenta posteriormente durante la fase de Ingeniería de la Aplicación. En esta fase se seleccionarán tanto las características del modelo de características como los atributos de calidad del modelo de calidad, con el fin de poder configurar un producto de acuerdo a determinados requerimientos funcionales y no funcionales. Las tres vistas del sistema (variabilidad, funcionalidad y calidad) serán las entradas de un plan de producción, que se servirá de las transformaciones de modelos para generar el producto final. En la Fig. 1 se muestran los metamodelos de las vistas de calidad y de variabilidad, las relaciones entre ellas y mediante las etiquetas (1) y (2) las relaciones con el metamodelo de la vista funcional. La vista de calidad, en sus clases característica, sub-característica y atributo contiene la información relativa a la perspectiva de calidad. Con ello podemos representar las diferentes perspectivas de calidad (interna, externa, calidad en uso etc.). Entre la vista de variabilidad y la vista de calidad tenemos una relación, expresada mediante la meta-clase *Impact* que, mediante un peso, representa la importancia relativa de dicha relación. Para representar la vista de variabilidad hemos empleado el metamodelo presentado en [4] que además nos permite introducir restricciones más complejas sobre el modelo.

Para la vista funcional hemos empleado como lenguaje de modelado SysML [10], dada su amplia difusión como herramienta de modelado en entornos MBSE en el dominio de los sistemas críticos. Concretamente para expresar la vista funcional empleamos el diagrama de bloques (BDD). En [9] se ilustra cómo se pueden distinguir atributos de calidad a nivel de *core asset*, y como estos pueden influir en atributos de calidad en el producto final, es por ello que se hace necesario relacionar la vista funcional y el modelo de calidad.



**Fig. 1** Relaciones entre la Vista de Calidad y la Vista de Variabilidad

La Fig. 2 muestra cómo expresamos estas relaciones entre la vista funcional y la vista de variabilidad y la de calidad. Entre estas vistas tenemos por un lado la relación entre una característica y un/os bloque/s funcional/es, para representar como una determinada característica puede relacionarse con uno o más bloques funcionales, y en caso de seleccionarse aquella, los bloques funcionales deberán estar presentes en el despliegue del producto. Además tenemos la relación entre un bloque funcional y atributo de calidad, mediante la meta-clase *Impact* que, que asocia un peso a dicha relación, representando la importancia relativa de dicha relación. Todos los pesos asociados en la clase *Impact* pueden ser tanto positivos como negativos, dependiendo de la naturaleza de dicha relación.



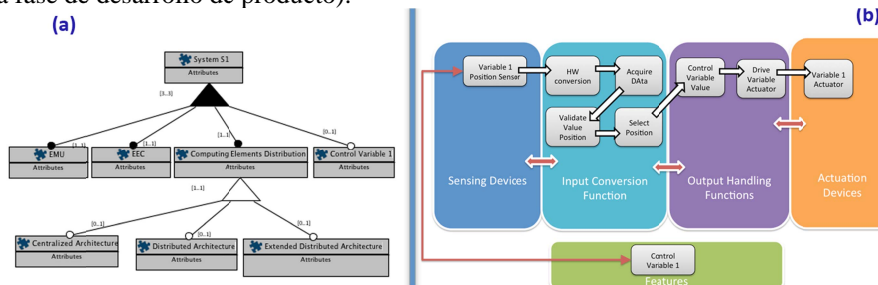
**Fig. 2.** Vista funcional y sus relaciones con las otras vistas del sistema

### 3. Aplicación de la propuesta

Las líneas de producto son muy atractivas en dominios tales como los sistemas críticos [8], dado que permiten reducir los costes de desarrollo. En este tipo de

entornos el tiempo de lanzamiento del producto no es tan importante como la adecuación al calendario con unos costes mínimos. La Fig. 3(a) muestra un extracto del modelo de características en los que se muestran algunas características opcionales en un sistema de control en vuelo (FADEC). Entre las características opcionales se encuentra la ubicación distribuida/centralizada de los nodos de computación [3]. El uso de múltiples nodos de computación es una característica que está empezando a explotarse e introduce mejoras sustanciales en atributos de calidad como seguridad, peso, coste etc.

La Fig. 3 (b) muestra el mapeado de una característica opcional en distintos bloques funcionales, y como estas se ubican en distintas capas de la arquitectura de la aplicación de control. En este caso se trata de una característica genérica que representa el control de posición de un determinado elemento mecánico, para ello son necesarios varios componentes software (bloques) como son la captura, conversión, validación, selección de la posición de este elemento mecánico, el cálculo del valor deseado y el control sobre el actuador para conseguir alcanzar dicho valor de posición. Las relaciones entre las características y los bloques funcionales puede ser en sí mismo una fuente de variabilidad, y ser tratado de nuevo como un punto de variación, que tendrá de nuevo impacto en los atributos de calidad del producto. Además los distintos bloques funcionales pueden tener en sí mismo impacto en los atributos de calidad tal como se analiza en [9], donde se describen varios atributos de calidad que impactan en la calidad del producto final y que pueden ser medidos a nivel de *core asset* o a nivel de producto final (en la fase de desarrollo de activos y en la fase de desarrollo de producto).



**Fig. 3.** Modelo de Variabilidad y Mapeado de los Bloques Funcionales

La Tabla 1 muestra la relación entre las características y los distintos atributos de calidad (peso, coste total, y coste de mantenimiento del sistema). Esta relación está expresada mediante valores relativos extraídos de las conclusiones recogidas en [3].

**Tabla 1.** Relación entre las características y los atributos de calidad

Features/Quality Attributes	Weight	System Cost	Maintenance Cost
Centralized Architecture	20	10	20
Distributed Architecture	60	50	40
Extended Distributed Architecture	80	70	80
Control Variable 1	10	5	5

## 4. Conclusiones y trabajos futuros

En el presente trabajo se ha presentado una aproximación basada en multi-modelos que nos permite introducir la perspectiva de calidad como una vista más del sistema, permitiendo además establecer relaciones entre el modelo de características y el modelo de calidad, y las relaciones existentes entre la vista de variabilidad y la vista funcional del sistema.

Como trabajos futuros pretendemos integrar el multi-modelo en procesos de transformación de modelos, que nos permitan seleccionar aquellas características que dotan al producto de unos determinados atributos de calidad. Evaluar, mediante técnicas de *model-checking*, si una configuración de características y atributos de calidad sería o no factible. Por último, y como siguiente paso, planteamos el diseño de planes de producción conducidos por el multi-modelo para sistemas críticos en entornos MBSE.

**Agradecimientos:** Este trabajo ha sido cofinanciado por el Ministerio de Ciencia e Innovación, en el marco del proyecto MULTIPLE (Ref.: TIN2009-13838) y el programa ValI+D de la Consellería de Educación y Ciencia (Generalitat Valenciana) Ref.: ACIF/2011/235

## Referencias

1. Boronat, A., Knapp, A., Meseguer, J. Wirsing, M.: What is a Multi-Modeling Language? In: LNCS, vol 5486/2009, pp 71—87 Springer, Heidelberg, (2009)
2. Clements, P., Northrop, N.: Software Product Lines: Practices and Patterns. Addison-Wesley, Boston (2001)
3. Culley, D., Palzewski, P. J., Storey, W., Smith, B. J.: The Case for Distributed Engine Control in Turbo-Shaft Engine Systems. In 65th Annual Forum and Technology Display, American Helicopter Society, Grapevine, Texas (2009)
4. Gómez, A., Ramos, I.: Cardinality-based feature modeling and model-driven engineering: Fitting them together. In Fourth International Workshop on Variability Modeling of Software intensive Systems (VAMOS'10), Linz (2010).
5. Gonzalez-Huerta, J., Blanes, J., Insfran, E., Abrahão, S.: Towards an Architecture for Ensuring Product Quality in Model-Driven Software Development. In: 11th International Conference on Product-Focused Software Process Improvement (PROFES) Limerick (2010)
6. Gonzalez-Huerta, J., Insfran, E., Abrahão, S.: Automatización de la Selección de Transformaciones Alternativas Basada en Atributos de Calidad. En Actas del VII Taller sobre Desarrollo de Software Dirigido por Modelos, pp. 10-18, Valencia (2010)
7. Insfran, E., Gonzalez-Huerta, J., Abrahão, S.: Design Guidelines for the Development of Quality-Driven Model Transformations. In: Proceedings of 13<sup>th</sup> International Conference on Model Drive Engineering Languages and Systems (MODELS2010), Part II, LNCS 6395, pp. 288–302, Oslo (2010)
8. Hutchesson, S., McDermid, J.: Development of High-Integrity Software Product Lines Using Model Transformation. In SAFECOMP, 2010.
9. Nolan, J.A., Abrahão, S., Clements, P., McGregor, J.D., Cohen, S.: Towards the Integration of Quality Attributes into a Software Product Line Cost Model. In 15th Software Product Line Conference, Munich (2011)
10. OMG: SysML 1.2 Specification: <http://www.omg.org/spec/SysML/1.2/PDF/>
11. Yoshimura, K., Forster, T., Muthig, D., Pech, D.: Model-Based Design of Product Line Components in the Automotive Domain. In 12th International Software Product Line Conference (SPLC2008), Limerick (2008)