

FMT: Una Herramienta de Modelado y Configuración de Líneas de Productos Software para MS Visual Studio*

Rubén Fernández, Miguel A. Laguna, Jesús Requejo, Nuria Serrano

Departamento de Informática, Universidad de Valladolid.
{luifern, mlaguna}@infor.uva.es, {jesus.requejo, nuriaserrano}@gmail.com

Resumen. Los modelos de características son instrumentos básicos para modelar, analizar y configurar la variabilidad de una línea de productos software. Todas estas tareas son complejas por lo que es necesario disponer de herramientas que satisfagan una serie de requisitos mínimos: generación automática de modelos, configuración de productos finales, integración en el entorno de desarrollo, etc. Por esa razón, hemos desarrollado una herramienta para realizar estas tareas, definiendo un lenguaje específico de dominio e implementando el mismo mediante las herramientas proporcionadas por Microsoft para Visual Studio. De esta forma hemos logrado integrar la herramienta en la misma plataforma utilizada para desarrollar la línea de productos, con lo que todas las fases de desarrollo comparten el mismo entorno.

1 Introducción

Uno de los problemas en el desarrollo de una línea de productos software (LPS) es la representación y gestión de la parte común y variable de la misma. La forma habitual de definir ambos aspectos es mediante modelos de características o features que permiten seleccionar la configuración de cada aplicación concreta dentro de la LPS. En la mayoría de LPS esas tareas son demasiado complejas para realizarlas manualmente. Por otro lado, en [3] hemos mostrado cómo se puede generar automáticamente la arquitectura básica de una línea de productos mediante transformaciones de los diagramas de características. La configuración de paquetes se basa en el mecanismo de *package merge* de UML 2 [4] y su implementación mediante clases parciales de C#. Hasta el momento para realizar todas estas tareas el grupo GIRO ha utilizado una combinación de herramientas: el *Feature Model Plug-in para Eclipse* de Czarnecki [1], y MS Visual Studio. La conexión entre ellas se realiza de manera manual utilizando ficheros XML y transformaciones con hojas de estilo XSLT. Los modelos de características se pueden convertir en la estructura de paquetes de una línea de productos, obtenida como un fichero XMI, que se puede importar con una herramienta CASE [3]. Actualmente existen diferentes herramientas libres que permiten la gestión de líneas de productos, como por ejemplo el mencionado *plug-in fmp* o Moskitt¹. Todas ellas permiten la definición de modelos de

* trabajo financiado por la Junta de Castilla y León (VA018A07) y el MICIINN (TIN2008-05675)

¹ <http://www.moskitt.org/eng/>

características así como su configuración y gestión de la variabilidad. Sin embargo ninguna de ellas proporciona la funcionalidad que deseamos en una herramienta de estas características, como transformación automática de modelos, integración en el entorno de desarrollo que utilizamos o generación de las aplicaciones finales de la línea de productos.

Por esto, hemos desarrollado una herramienta, Feature Modeling Tool² (FMT), que nos permita realizar diagramas de características, así como gestionar la variabilidad, realizar configuraciones de productos específicos y además integrar todo ello en MS Visual Studio. Nuestra motivación es disponer de una herramienta que integre todas las fases del proceso en el mismo entorno de desarrollo y realice de manera automática las tareas que hasta el momento se hacen exportando e importado los distintos modelos de manera manual. En el resto del documento se describe la herramienta con más detalle y se muestra un ejemplo de uso.

2 Descripción de la Herramienta FMT

Para definir los modelos de características que crearemos con FMT hemos seleccionado el meta-modelo que utiliza Czarnecki [1] en el *plug-in* para Eclipse. Hemos elegido este meta-modelo porque es uno de los más utilizados y para facilitar la compatibilidad con *fmp*, permitiendo la importación/exportación de modelos.

Para desarrollar FMT y poder integrarla en el mismo entorno de desarrollo que utilizaremos para desarrollar los productos, hemos utilizado DSL Tools [2]. DSL Tools es una herramienta incluida en el SDK de MS Visual Studio que permite definir lenguajes de dominio específico y crear herramientas de modelado para esos lenguajes. Entre las ventajas que aporta FMT podemos destacar la representación gráfica de los modelos de características incluyendo restricciones tipo *mutex/require* y sus posibilidades de automatización: transformación de modelos y generación de las aplicaciones finales de la LPS. La Figura 1 muestra la interfaz de FMT.

La ventana principal se divide en tres secciones. La ventana central es el panel de diseño, donde podemos crear los diagramas de características. La visión gráfica se complementa con el explorador en árbol, idéntico al formato utilizado por *fmp* (panel inferior derecho). La sección de la izquierda es el panel de configuración. Desde este panel podemos seleccionar aquellas características que queremos que formen parte del producto específico a desarrollar en cada caso. La propia herramienta se encarga de comprobar las multiplicidades y otras restricciones como las relaciones *require* y *mutex*. Por último, en la tercera sección de la ventana localizada en la parte derecha tenemos la configuración de la LPS como un proyecto de desarrollo (una solución de Visual Studio). En esta ventana podemos ver la transformación en el modelo de paquetes para nuestra configuración del diagrama de características. Si marcamos una característica en el panel de configuración su paquete correspondiente será añadido a la solución.

Otra función de la herramienta es la importación/exportación de modelos. Podemos importar un modelo creado previamente con el *plug-in fmp*. También

² disponible en <http://giro.infor.uva.es/FeatureTool.html>

podemos exportar nuestro modelo de características a un modelo UML (en formato XMI). Para ello la herramienta realiza la transformación propuesta en [3], utilizando las facilidades de manejo de ficheros XML proporcionados por la plataforma. Otra de las funciones permite la exportación en el formato de la herramienta FaMa [5] que realiza la validación de los modelos de características.

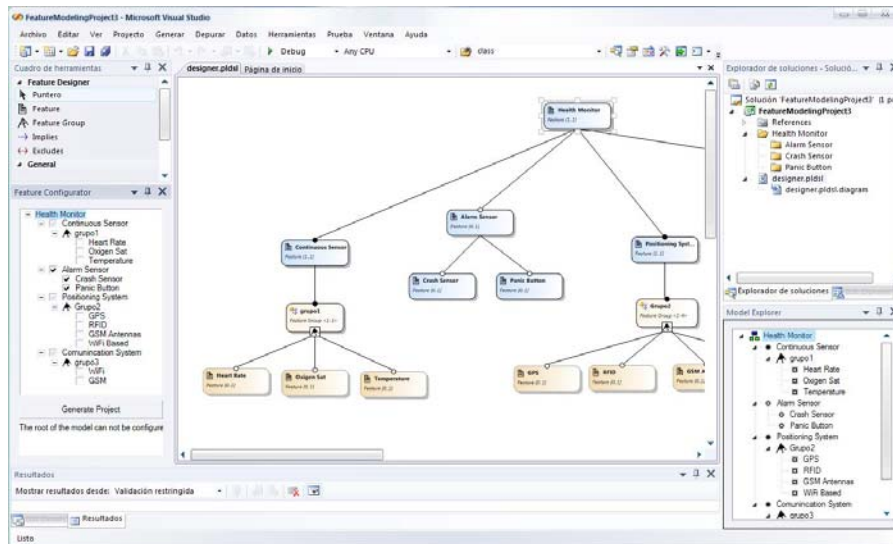


Fig. 1 Interfaz de FMT

3 Ejemplo de Uso

Para mostrar el funcionamiento de la herramienta, la Figura 2a muestra el flujo de trabajo a seguir en el desarrollo de los activos de la LPS. Inicialmente se crea el modelo de características. Podemos hacer esto de dos maneras: bien creándolo manualmente en la ventana de diseño o bien importando un modelo ya existente desde el *plug-in fmp*. Ahora podemos hacer cualquier modificación que creamos conveniente sobre este modelo. Podemos añadir restricciones gráficas que el *plug-in fmp* no permite, como relaciones *require* y *mutex* o asignar un tipo a las características hoja. Una vez finalizado el diagrama de características de nuestra línea de productos, el siguiente paso es crear la solución de Visual Studio para comenzar el desarrollo. FMT crea automáticamente la solución generando los paquetes correspondientes a la transformación del modelo de características. Si seleccionamos la opción “*export model*”, la herramienta realiza la transformación del modelo de características a un modelo de paquetes UML.

El siguiente paso consiste en completar esos paquetes. Una vez que los paquetes de la LPS están completos podemos configurar aquellos productos finales que deseamos. Basta con seleccionar en el panel de configuración aquellas características que queremos que formen parte del producto y Visual Studio seleccionará los paquetes necesarios y compilará la solución (Figura 2b).

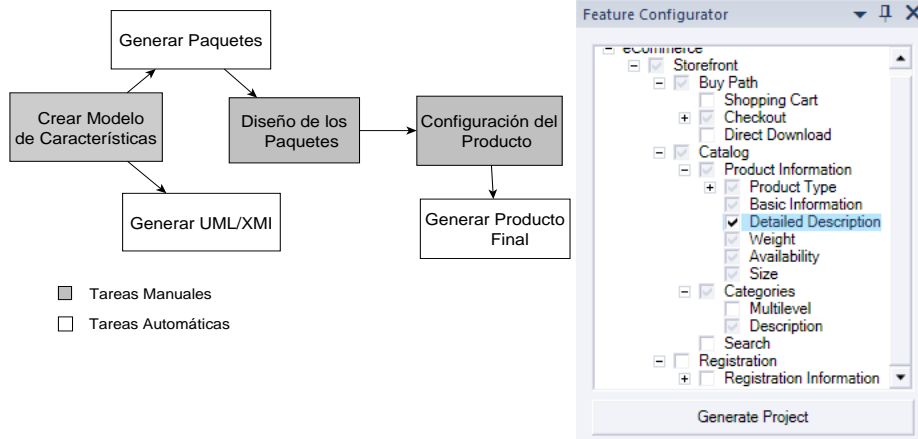


Fig. 2 Flujo de trabajo para el desarrollo de una LPS (a) y herramienta de configuración (b)

4 Conclusiones

Como resumen final, hemos desarrollado una herramienta que no solo nos permite gestionar y configurar diagramas de características, sino que también realiza de forma automática las transformaciones entre los diferentes modelos, creando la arquitectura básica de la línea de producto, automatizando la mayoría de las tareas del proceso de desarrollo e integrando toda la funcionalidad en la misma herramienta que será utilizada posteriormente para la generación de los productos finales de la LPS. Se encuentra en fase de integración la generación automática de interfaces de usuario a partir de la combinación de archivos XML parciales.

Referencias

1. M. Antkiewicz and K. Czarnecki. "Feature Plugin: Feature modeling plug-in for Eclipse." In OOPSLA'04 Eclipse Technology eXchange (ETX) Workshop, 2004.
2. Steve Cook, Gareth Jones, Stuart Kent, Alan Cameron Wills. "Domain-Specific Development with Visual Studio DSL Tools (Microsoft .NET Development Series)" Junio 2007. Addison-Wesley Professional
3. Miguel A. Laguna, Bruno González-Baixauli. "Variabilidad, Trazabilidad y Líneas de Productos: una Propuesta basada en UML y Clases Parciales." JISBD 2007, Zaragoza, Spain - sep 2007
4. Object Management OMG. "Unified modeling language specification version 2.0: Infrastructure." Technical Report ptc/03-09-15, OMG, 2003.
5. Pablo Trinidad, David Benavides, Antonio Ruiz Cortés, Sergio Segura, Alberto Jiménez. FAMA Framework. SPLC 2008: 359