

Configuración de productos de una línea de producto software utilizando un lenguaje específico del dominio

David de Castro, Alejandro Cortiñas
Miguel R. Luaces y Óscar Pedreira

Universidade da Coruña, Centro de Investigación CITIC, Laboratorio de Bases de Datos, Facultade de Informática, Elviña, 15071 A Coruña, Spain
{david.decastro,alejandro.cortinas, luaces,opedreira}@udc.es

Resumen La selección de una característica en una Línea de Producto Software determina si esta se encontrará presente o no en el producto final. Sin embargo, en nuestra experiencia trabajando con tecnologías LPS en el ámbito de los Sistemas de Información Geográfica, hemos identificado escenarios en los que puede ser necesario aplicar una característica a un elemento concreto de la aplicación y no a todo su conjunto. Es por ello por lo que en esta publicación se plantea una solución que, haciendo uso de Lenguajes Específicos de Dominio, permita asociar características con determinados elementos de la aplicación generada, con el fin de alcanzar una mayor personalización del SIG generado y mejorando la calidad de estos.

Keywords: Lenguaje específico de dominio · Sistemas de Información Geográfica. · Línea de Producto Software

1. Introducción

En el Laboratorio de Bases de Datos trabajamos desde hace años con una Línea de Producto Software (LPS) orientada a la generación de Sistemas de Información Geográfica (SIG), aplicaciones web que permiten la visualización y gestión de datos geográficos de diversa índole. Para poder generar un SIG con la LPS se define una especificación que cuenta principalmente con dos elementos: un conjunto de características que se desea incluir, y el modelo de datos que representa el dominio de la aplicación en cuestión.

En la primera versión de dicha LPS esta especificación era definida a través de una aplicación web, desde la cuál se seleccionaban las características y se definía el modelo. Pero esta aplicación no permitía configurar determinadas partes del producto a generar, como pueden ser los mapas, capas y estilos. Es por ello que, posteriormente, se definió un Lenguaje Específico de Dominio (conocido por sus siglas en inglés, DSL) [2] que permitía la definición de estos, logrando así un mayor nivel de personalización, y se integró con la LPS. De esta forma, la

especificación de un producto incluía tanto las características seleccionadas y el modelo de datos como la definición del conjunto de mapas, capas y estilos con las que debe contar la aplicación.

Sin embargo, si bien el DSL permite crear productos más personalizables, el hecho de que la selección de características se aplique de forma global sigue suponiendo un problema, especialmente cuando generamos SIG complejos o grandes. Un caso real con el que existen estas limitaciones es WebEIEL [1], una aplicación web dedicada a servir contenidos relativos a la realidad geográfica de la provincia de A Coruña, su computación, análisis, estudio y representación cartográfica. En este producto existen determinadas vistas de mapas cuyos datos deben visualizarse “clusterizados” (es decir, agrupados por cercanía) para que el mapa sea manejable por la cantidad de datos. Sin embargo, otros mapas más simples con menos datos deberían verse sin “clusterizar”, dado que se necesita ver los datos individuales. La selección de determinadas características de manera global provoca la imposibilidad de personalizar estas vistas de manera simultánea.

En este artículo corto presentamos la solución al problema descrito, que pasa por alterar la naturaleza de algunas características de nuestro modelo, pasando a formar parte del DSL y afectando a la aplicación de manera local. Este cambio aumenta el nivel de personalización admitida por la LPS, y permite personalizar determinadas funcionalidades así como enlazar características con entidades concretas del modelo de datos.

2. Nuestra propuesta

En la Figura 1 se puede ver un fragmento del árbol de características original de la LPS. Algunas de las funcionalidades que se pueden seleccionar son: *LayerManager*, que sirve para disponer de un gestor de capas en el cuál se dispondrán los selectores de estilo, opacidad, etc. dependiendo de si las correspondientes características están activadas; *Clustering*, que sirve para agrupar elementos del mapa que se encuentran muy cerca para evitar solapamientos; u *OpacitySelector*, que añade la funcionalidad en el *LayerManager* de cambiar la opacidad de una capa. Hasta ahora, si se selecciona la característica *LayerManager*, todos los mapas de la aplicación pasan a tener un selector de capas, independientemente de que realmente lo necesitaran. Lo mismo sucede con la *clusterización*, ya comentada en la introducción del artículo, o con la funcionalidad *OpacitySelector*, entre otras.

Tomando como base un DSL, en la Figura 2 se puede ver una parte del modelo de este, donde se pueden definir mapas de la aplicación, seleccionando para cada mapa el conjunto de capas que lo componen. Asociado a la entidad *Map* ahora vemos el árbol de características *MapFeatures*, que representa las características que se pueden aplicar para cada mapa de la aplicación. Del mismo modo tenemos el modelo *LayerFeatures*, asociado en este caso a cada capa.

Para que la LPS soporte este cambio, se ha tenido que adaptar el código de forma que no tenga únicamente en cuenta las características globales seleccionadas, sino también las características locales asociadas a cada elemento; para

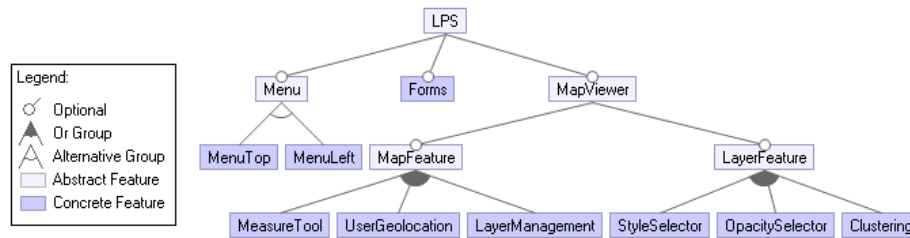


Figura 1: Árbol de características de la LPS

lograr esto, fue necesario generar en el código unos mecanismos de control que únicamente permitan acceder al usuario a determinadas características cuando la instancia de la clase lo permita; es decir, que tenga asociada la característica en concreto. De esta forma se consigue la correcta integración de las características con los elementos de la aplicación, alcanzando una mayor personalización del producto final. Esto, unido con la posibilidad de definir múltiples mapas diferentes (eliminando así el único mapa que contiene todas las entidades geográficas del producto) se logra una mejora significativa del rendimiento y un nivel de personalización mucho mayor del alcanzado hasta el momento.

Tomando como ejemplo el producto WebEIEL presentado en la introducción, esta solución permite solucionar los problemas revelados. Por una parte, es posible definir múltiples mapas, por lo que el rendimiento de estos mejora de forma notable ya que no se encuentran todos los elementos geográficos insertados en un mismo mapa. Por ejemplo: tener un mapa para visualizar el índice de cultura de los municipios (con diferentes colores dependiendo del nivel) y otro mapa para visualizar todos los parques de la provincia. Pero con los cambios planteados, también se abre la posibilidad de asociar características a elementos concretos, como pueden ser las capas, eliminando así funcionalidades innecesarias de aquellas que no las requieren. Siguiendo el ejemplo anterior, la característica de agrupación de elementos de una capa en clústers tiene sentido aplicarla a la capa de parques, ya que son muchos puntos en el mapa, muy juntos entre sí y que se solapan, pero carece de sentido en la capa de municipios, ya que aquí todos los elementos geográficos no se llegan a superponer nunca.

Con estos cambios se logra ese mayor nivel de personalización propuesto, se simplifica el desarrollo de SIG complejos y se mejora su rendimiento.

3. Conclusiones y trabajo futuro

La combinación de tecnologías de Líneas de Producto Software e Ingeniería Dirigida por Modelos permite generar de forma semi-automática Sistemas de Información geográfica con funcionalidades y características básicas, aunque todavía cuenta con ciertos puntos que necesitan ser mejorados para alcanzar un nivel de madurez que permita alcanzar un mayor grado de personalización de

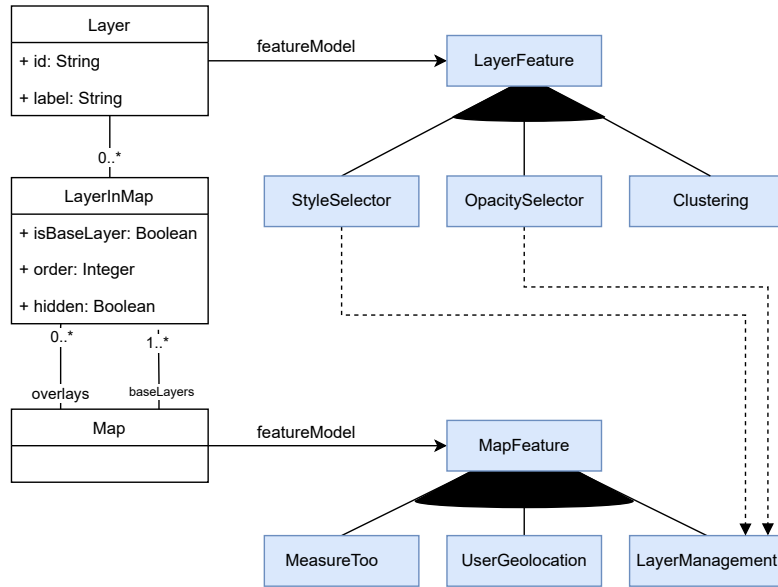


Figura 2: Ejemplo de integración de características en el modelo DSL

los productos generados. Los cambios propuestos en esta publicación solucionan esos puntos y ayudan a crear software de una mayor calidad.

Como trabajo futuro se plantea la realización de todos los cambios propuestos en esta publicación. Sin embargo, es necesario valorar la efectividad de estos en un contexto real, por lo que otro de los objetivos marcados a futuro es la puesta en funcionamiento de la solución desarrollada para la generación de un producto real, como puede ser la WebEIEL (mencionada previamente), lo que nos permitirá comprobar la efectividad real de estos cambios.

Agradecimientos Parcialmente financiado por: MCIN/AEI/10.13039/50110-0011033, NextGenerationEU/PRTR, FLATCITY-POC: PDC2021-121239-C31; MCIN/AEI/10.13039/501100011033 EXTRACompact: PID2020-114635RB-I00; GAIN/Xunta de Galicia/ERDF CEDCOVID: COV20/00604; Xunta de Galicia/FEDER-UE GRC: ED431C 2021/53; MICIU/FEDER-UE BIZDEVOPSGLOBAL: RTI-2018-098309-B-C32; MCIN/AEI/10.13039/501100011033 MAGIST: PID2019-105221RB-C41.

Referencias

1. WebEIEL. <https://webeiel.citic.udc.es/>
2. Alvarado, S.H., Cortiñas, A., Luaces, M.R., Pedreira, O., Places, A.S.: Developing web-based geographic information systems with a dsl: proposal and case study. *Journal of Web Engineering* (2020)