

Un Lenguaje específico de dominio para el desarrollo de aplicaciones domóticas

M^a Francisca Rosique, Manuel Jiménez, Pedro Sánchez, Bárbara Álvarez, Andrés Iborra
 División de Sistemas e Ingeniería Electrónica (DSIE)
 Universidad Politécnica de Cartagena, 30202 Cartagena, España
paqui.rosique@upct.es | manuel.jimenez@upct.es | pedro.sanchez@upct.es

Resumen. *Los sistemas domóticos han surgido como uno de los campos de mayor interés en el ámbito de la ingeniería dada la actual demanda de los sistemas de la información en la sociedad. Actualmente el desarrollo de estos sistemas se realiza en un espacio muy cercano a la solución, de forma dependiente de la plataforma y a un bajo nivel de abstracción. En este sentido, el lenguaje específico de dominio (DSL) que se presenta en este trabajo facilita la captura de requisitos propios de un sistema domótico de forma visual e intuitiva.*

1. Introducción

Los lenguajes visuales transmiten mayor información por unidad de expresión, ya que los elementos gráficos ayudan al entendimiento, son fáciles de identificar y recordar y son lenguajes entendibles de forma universal [1].

Para el caso concreto del dominio domótico sería deseable disponer de un lenguaje visual que a la vez sea específico para este dominio. La definición de este lenguaje tiene por objetivo ayudar a los diseñadores a describir los sistemas domóticos utilizando únicamente conceptos del dominio. En este sentido, el DSL que se presenta en este trabajo facilita la captura de requisitos propios de un sistema domótico de forma visual e intuitiva. De esta forma, los usuarios ven facilitada la posibilidad de expresar y entender su conocimiento y experiencia en el dominio. Por esta razón la primera premisa es la de disponer de una riqueza semántica importante para la visualización del conocimiento, pero a la vez concisa y común a las distintas plataformas. Antes de entrar en detalles es necesario exponer los principales conceptos con los que se trabaja en el dominio domótico y que deben tenerse en cuenta a la hora de crear el DSL.

En cualquier sistema domótico [2] existe una serie de elementos (que denominamos "Unidades Funcionales") que aparecen en todas las tecnologías y estándares domóticos. Se diferencian en la arquitectura, protocolos utilizados o módulos disponibles, pero son iguales en cuanto funcionalidad. Con el fin de promover la reutilización de estas unidades funcionales y evitar tener que definir múltiples veces la misma unidad para cada aplicación (incluso varias veces dentro de una misma aplicación), se ha optado por utilizar un Catálogo de unidades funcionales reutilizable, de manera que una vez definido dicho catálogo éste se pueda utilizar en cualquier aplicación y sólo sea necesario obtener ejemplares de dicho catálogo. Estas unidades funcionales a su vez disponen de unos Servicios gracias a los cuales las unidades

podrán interactuar con otras unidades. Muchos de estos servicios se repiten entre las unidades funcionales, de manera que se ha creado un Catálogo de Servicios con unas Definiciones de Servicios que puedan ser reutilizadas en cualquier unidad funcional.

Por esta razón es conveniente distinguir dos vistas del DSL, una para el desarrollo de aplicaciones con un desarrollador de aplicaciones como usuario (conoce el dominio pero no tiene que ser experto) y una segunda vista para desarrollar y realizar posibles actualizaciones del catálogo, donde el usuario deber ser un experto en el dominio.

1.1 Vista Catálogo del DSL

El DSL permite al experto en el dominio modelar el catálogo de unidades funcionales y servicios. Este catálogo será utilizado posteriormente para el desarrollo de aplicaciones domóticas, sin que sea necesario definir cada una de las unidades funcionales y sus servicios cada vez que se desarrolle una nueva aplicación.

Las primitivas disponibles para modelar un catálogo se describen en la parte superior de la Tabla 1.

Un catálogo incluye categorías y unidades funcionales. Las categorías forman una jerarquía estática definida siguiendo una notación cercana al dominio domótico. En el catálogo desarrollado se realizan dos agrupaciones principales, una para los elementos "Final-Passive" (aquellos puramente hardware) y otra agrupación para los "Controllers" que son elementos que aportan una funcionalidad configurable por software. En un segundo nivel se distinguen entre elementos de entrada, salida o elementos de entrada/salida.

La actualización y el incremento de dicho catálogo se centra en la incorporación de nuevas definiciones de unidades funcionales dentro de las categorías existentes.

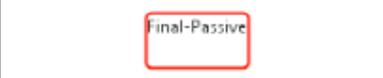
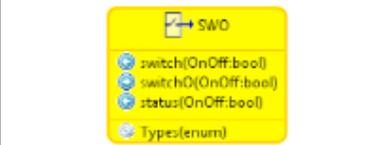
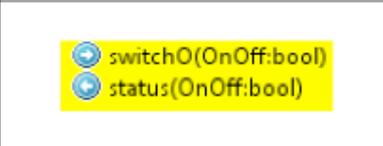
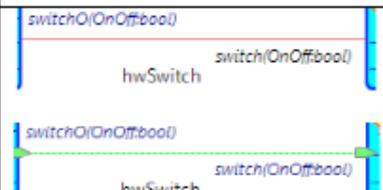
Representación gráfica y descripción de las primitivas de la vista Catálogo	
	Categoría: Especialización de un elemento del catálogo.
	Enlaces entre categorías (verde) y enlaces entre unidades funcionales y categorías (azul).
	Definición de Unidad Funcional: es el mínimo elemento en el que puede dividirse un dispositivo doméstico. Incluye un icono, un nombre y los servicios provistos e implementados..
	Servicio: Una definición de servicio tiene una firma que incluye el nombre del servicio y sus argumentos. El servicio indica si es provisto o requerido (flecha izq o drcha) e indica si es un servicio hardware o no (flecha roja o azul). Un catálogo de servicios se usa para permitir la reutilización de los servicios.
	Definición de parámetro: indica el parámetro de una unidad funcional.
	Definición de escena: es una especialización de unidad funcional.
Representación gráfica y descripción de las primitivas de la vista Aplicación	
	Instancia de unidad funcional: incluye los parámetros y sus valores.
	Enlace entre unidades funcionales. Un enlace del tipo canal se muestra en rojo, el resto de enlaces se muestran con una línea discontinua verde. Estos enlaces tienen tres etiquetas con los nombres de los servicios involucrados en dicho enlace.
	Escena: contiene los pasos de escena que se ejecutarán. Un paso de escena muestra el servicio y el icono de la unidad funcional que participa en dicho paso de escena..

Tabla 1: Elementos de la vista Catálogo y Aplicación.

1.2 Vista Aplicación del DSL

Esta vista del DSL será utilizada por el desarrollador, encargado de definir/diseñar nuevas aplicaciones y sin tener que ser un experto en el dominio doméstico. En este sentido, y gracias a la existencia del catálogo, la especificación de una aplicación doméstica se realiza mediante:

- La instanciación de unidades funcionales que vienen definidas en el catálogo. Estas instancias de unidades funcionales deben ser configuradas añadiendo los valores necesarios a sus parámetros. Por ejemplo, para una unidad funcional definida como DMI que tiene un parámetro DMIlongPushTime del tipo numérico se deberá indicar el valor que debe tomar dicho parámetro. Se podría dar un valor para que

pasados 3 segundos de pulsación detecte una pulsación larga y empiece a regular, en caso contrario realizará un conmutación.

- Los enlaces entre unidades funcionales. Mediante enlaces se puede indicar la forma en la que las unidades funcionales van a interactuar con el resto del sistema. Al realizar un enlace se indican los servicios que se interconectan. Estos enlaces pueden comportarse como enlaces de tipo canal en el caso de que una de las unidades funcionales involucradas sea pasiva de manera que se modela una conexión hardware o bien como un enlace normal.
- Escenas. Se puede configurar la ejecución de varios servicios de unidades funcionales de forma secuencial con una única acción. Por ejemplo se puede crear una escena

"Presentación" que se activa desde un pulsador y ejecuta una serie de servicios preestablecidos con antelación, por ejemplo bajar persiana, bajar intensidad de las luces y bajar pantalla de proyección.

Las primitivas disponibles en esta vista de aplicación son las que se muestran en la parte inferior de la Tabla 1.

2. Herramienta DSL y entorno de trabajo

El DSL se ejecuta en un entorno Eclipse, por lo que mantiene una estructura visual muy parecida con el resto de herramientas de Eclipse. En la figura 1 se pueden observar 3 zonas bien diferenciadas:

- **Área de dibujo:** zona de trabajo donde se arrastran las figuras desde la paleta para construir un modelo gráfico concreto, tanto para la vista de Catálogo como de Aplicaciones. También existe la opción de instanciar elementos desde un menú flotante en lugar de utilizar la paleta.
- **Paleta gráfica:** contiene las primitivas que se pueden arrastrar a la zona de dibujo.
- **Área de propiedades:** zona donde se visualizan y se modifican aquellas propiedades (atributos, parámetros, etc...) disponibles para la primitiva seleccionada. Para acceder a estas propiedades basta con seleccionar un elemento instanciado en la zona de dibujo. Las propiedades se muestran como campos editables o desplegables para facilitar un acceso rápido e intuitivo.

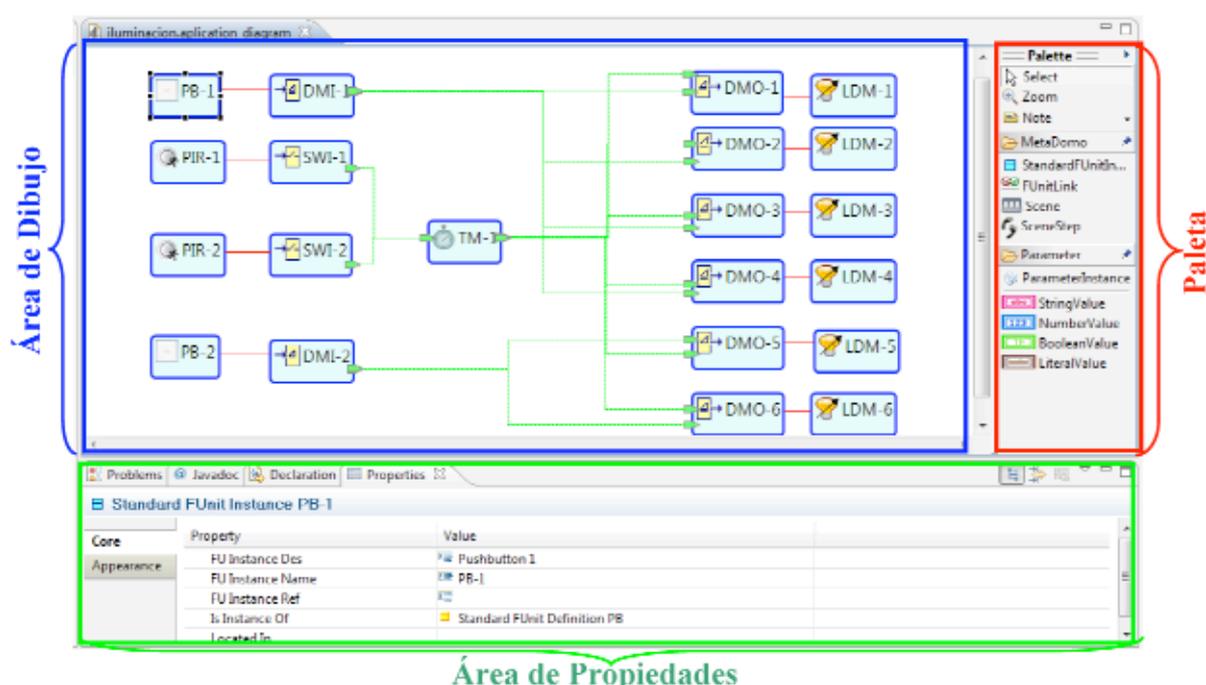


Figura 1: Vista Aplicación del DSL .

3. Conclusiones

La utilización del DSL propuesto en este documento proporciona al usuario (que ahora no tiene que ser experto) un entorno gráfico con un alto nivel de abstracción donde definir los requisitos del sistema con conceptos propios del dominio domótico. De este modo no se hace necesario comenzar el desarrollo analizando el software desde una perspectiva tecnológica.

El beneficio obtenido en el dominio domótico es doble: por un lado se ha conseguido separar en dos vistas complementarias el catálogo de dispositivos (presentes o futuros) y el diseño de nuevas aplicaciones, por otro, al integrarse esta propuesta en un marco de desarrollo dirigido por modelos [3], se obtiene la infraestructura conceptual más adecuada

para beneficiarse de las posibilidades relativas a: generación automática de código, verificación y validación de modelos, trazabilidad de requisitos, etc.

Referencias

- [1] J. Memik M., Lämmel R., "Special issue on domain-specific languages, Part II.", Journal for Computing and Information Technology, 10(1), 200 Miller, J.
- [2] J.I. Ryan, "Home Automation", Electronics & Communication Engineering Journal, Volume: 1 Issue:4, July-Aug. 1989, pp 185-192
- [3] Rosique, MF et al., "Desarrollo de aplicaciones domóticas dirigido por modelos". I Jornadas de Introducción a la investigación de la UPCT". pp 6-8, 2008.