

# Reúso efectivo de modelos domóticos a través de requisitos genéricos

Francisca Rosique, Diego Alonso, Pedro Sánchez, Francisco Sánchez  
Departamento de Tecnologías de la Información y las Comunicaciones. ETSI Telecomunicaciones  
Campus Muralla del Mar 30202 Cartagena (Murcia)  
e-mail: paqui.rosique@upct.es

**Resumen.** *En el desarrollo de software para sistemas domóticos existen ciertos conceptos susceptibles de ser reutilizados en distintos desarrollos o proyectos. Esta reutilización debe aprovecharse desde las primeras etapas de desarrollo y es por ello que en este artículo se propone iniciar el desarrollo de una nueva aplicación a partir de la reutilización de requisitos existentes y previamente catalogados. Cada uno de estos requisitos ha sido asociado con su correspondiente especificación gráfica, representada a través de un Lenguaje Específico del Dominio (DSL). Es decir, cada requisito tiene asociado un fragmento de modelo DSL, de esta manera, quedan catalogadas las soluciones parciales a cada requisito domótico y al reutilizar los requisitos se reutiliza de forma indirecta fragmentos del modelo.*

## 1. Introducción

La Ingeniería de Requisitos es una disciplina en auge, que ha demostrado su capacidad de influencia en la mejora de la productividad y calidad en los procesos y productos software. Dentro de esta disciplina es de especial interés la reutilización de requisitos. El propósito de la reutilización de requisitos es identificar descripciones de sistemas que puedan ser reutilizadas (en su totalidad o en parte) con un mínimo número de modificaciones, de manera que se reduzca el esfuerzo total de desarrollo. Este nivel de reutilización puede aportar grandes beneficios.

Actualmente la forma más utilizada de representación de requisitos es el lenguaje natural, pero además de los problemas inherentes al lenguaje natural, se constata que la diversidad de formatos de requisitos es una restricción para su reutilización y por otro lado en se afirma que, cualquier diagrama o especificación que permita hacer los requisitos visibles incrementan la posibilidad de reutilización.

Con estas hipótesis de partida se ha considerado más que necesario que la fase de captura de requisitos se realice haciendo uso de modelos gráficos que permitan a su vez catalogar y almacenar los requisitos para su uso y reutilización en futuros desarrollos. Para ello es necesaria una herramienta de soporte para la gestión de requisitos en general.

Varios autores señalan la ausencia de soluciones prácticas a la reutilización de requisitos con herramientas comerciales y presentan guías para reutilizar y clasificar requisitos.

Siguiendo con estas indicaciones se ha creado una metodología de desarrollo que permite al usuario desarrollar una nueva aplicación a partir de la reutilización de requisitos existentes y previamente catalogados. Cada uno de estos requisitos ha sido asociado con su correspondiente especificación gráfica, representada a través de un Lenguaje

Específico del Dominio (DSL). Es decir, cada requisito tiene asociado un fragmento de DSL, de esta manera, quedan catalogadas las soluciones parciales a cada requisito domótico.

Un fragmento forma parte de un modelo completo en el sentido de que lleva a cabo parte de la funcionalidad deseada. Es posible que un fragmento sea sintáctica o semánticamente incompleto. Por lo tanto, las reglas de integridad deben estar desactivadas temporalmente con el fin de facilitar la integración de estos fragmentos en un modelo de aplicación.

Al construir una nueva aplicación, el usuario debe inspeccionar dicho catálogo de requisitos identificando los requisitos que va a reutilizar del mismo. Al hacerlo, reutiliza las soluciones adoptadas con el DSL en el desarrollo de aplicaciones anteriores. Por último, todos los fragmentos del DSL (reutilizados y nuevos), se integran en un único modelo sobre el cual se realizarán una serie de transformaciones que nos permitirán obtener la aplicación final.

## 2. Catálogo de requisitos

En el desarrollo de sistemas domóticos es habitual que los requisitos básicos estén repetidos o que sean muy parecidos entre distintos proyectos. Por esta razón se ha pensado en crear un catálogo que permita reutilizar requisitos entre los distintos sistemas domóticos desarrollados. En la literatura podemos encontrar múltiples alternativas para reutilización de requisitos, algunas de ellas incluyen sus propios metamodelos que permiten reflejar los conceptos relevantes para cada trabajo. Nosotros hemos optado por la sencillez y tomando algunos de los trabajos existentes como referencia se ha adoptado el siguiente metamodelo mostrado en la figura 1.

Este metamodelo, tiene una versatilidad suficiente con una estructura simple que podría ser aplicado también a otros dominios.

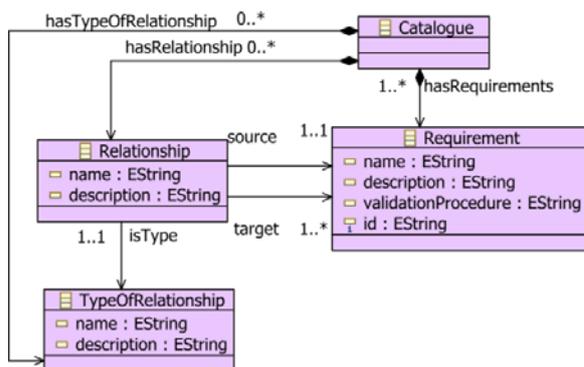


Figura 1. Metamodelo de requisitos adoptado

El elemento raíz es "Catalogue", donde se incluyen el conjunto de requisitos. Cada uno de estos requisitos incluye un nombre, una descripción y un procedimiento de validación para el usuario con el fin de validar si el sistema cumple la funcionalidad esperada.

Los requisitos normalmente se encuentran fuertemente relacionados entre ellos, por lo que se incorporó una EClass "Relationship" que permite relacionar varios requisitos entre sí. El tipo de relación entre requisitos se identifica con el elemento "TypeOfRelationship".

Para el caso de dominio domótico hemos creado un catálogo dividido en cuatro grandes categorías de requisitos: Confort-luces (iluminación), confort-motorización (para dispositivos móviles a través de motores, como por ejemplo persianas, pantallas de proyección, toldos, etc), confort-Climatización y por último, seguridad (para dispositivos como alarmas, envíos de mensajes, detectores, etc). Los requisitos de comunicación no han sido separados en una categoría específica dado que son transversales al resto de requisitos catalogados en las otras cuatro categorías.

Cada uno de los requisitos tiene asociado un fragmento de modelo DSL donde se representa en notación gráfica el requisito. Esta asociación se ha realizado de forma manual y la selección actual de los requisitos del catálogo también se realiza de forma manual, aunque se esta desarrollando (en colaboración con la Universidad de Murcia) un plugin en java que permitirá en un futuro poder realizar la selección de una forma automática.

De esta manera, a la hora de desarrollar una aplicación sólo se debe seleccionar los requisitos necesarios del catálogo y automáticamente se genera un modelo inicial de aplicación en notación DSL (véase Figura 3). Este modelo puede ser modificado posteriormente para refinar el modelado de la aplicación.

Cada una de las correspondencias entre requisito y elementos del DSL se han trazado con el fin de poder realizar un seguimiento de estos requisitos a lo largo del ciclo de vida de desarrollo del software para el

sistema domótico y poder gestionar un control de cambios exhaustivo.

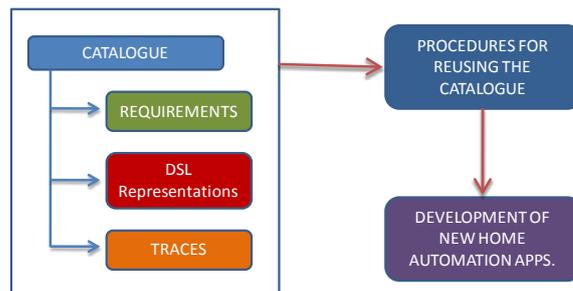


Figura 2. Esquema de los pasos necesarios para usar el catálogo de requisitos

## 4. Conclusiones

En este documento se presenta un enfoque que permite incrementar el reuso de modelos a través de requisitos genéricos. Para ello se ha creado un catálogo de requisitos domóticos cuya representación gráfica se corresponde con la definida en el DSL domótico Habitation. Estos representaciones gráficas se corresponden con fragmentos de modelos, de modo que se puede aprovechar la etapa de análisis de requisitos para crear una primera aproximación del modelo del sistema y de este modo integrar toda la etapa de requisitos en un framework domótico completo junto con las distintas herramientas creadas para cubrir las necesidades de cada fase del ciclo de vida. Estas herramientas incluyen (1) un DSL para la creación de modelos de requisitos domóticos, (2) un modelo intermedio de componentes para reducir la complejidad y facilitar la confluencia con otros dominios de sistemas reactivos, (3) modelos para las plataformas específicas de ejecución y (4) un conjunto de transformaciones para refinar los modelos de requisitos, pasando por los diferentes niveles de abstracción, hasta obtener código ejecutable. Y todo esto extendido con mecanismos de trazabilidad de todos los modelos involucrados en el proceso.

Como líneas futuras sería interesante contar con una herramienta que permita poder realizar la trazabilidad de forma automática de los requisitos con sus correspondientes elementos del fragmento de modelo gráfico definido con el DSL. Así como la composición del modelo gráfico de la aplicación. Actualmente estos procesos se realizan manualmente y pueden estar expuestos a posibles fallos humanos. Una posible forma de realizar esta automatización sería a través de transformaciones de modelos usando el lenguaje ATL por ejemplo, aunque al tratarse de modelos gráficos también se podría contemplar el uso de herramientas para transformación de grafos.

## Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto español de investigación EXPLORE (ref. TIN2009-08572) del CICYT y por el proyecto de investigación MISSION-SICUVA (ref. 15374/PI/10) del gobierno de la Región de Murcia.

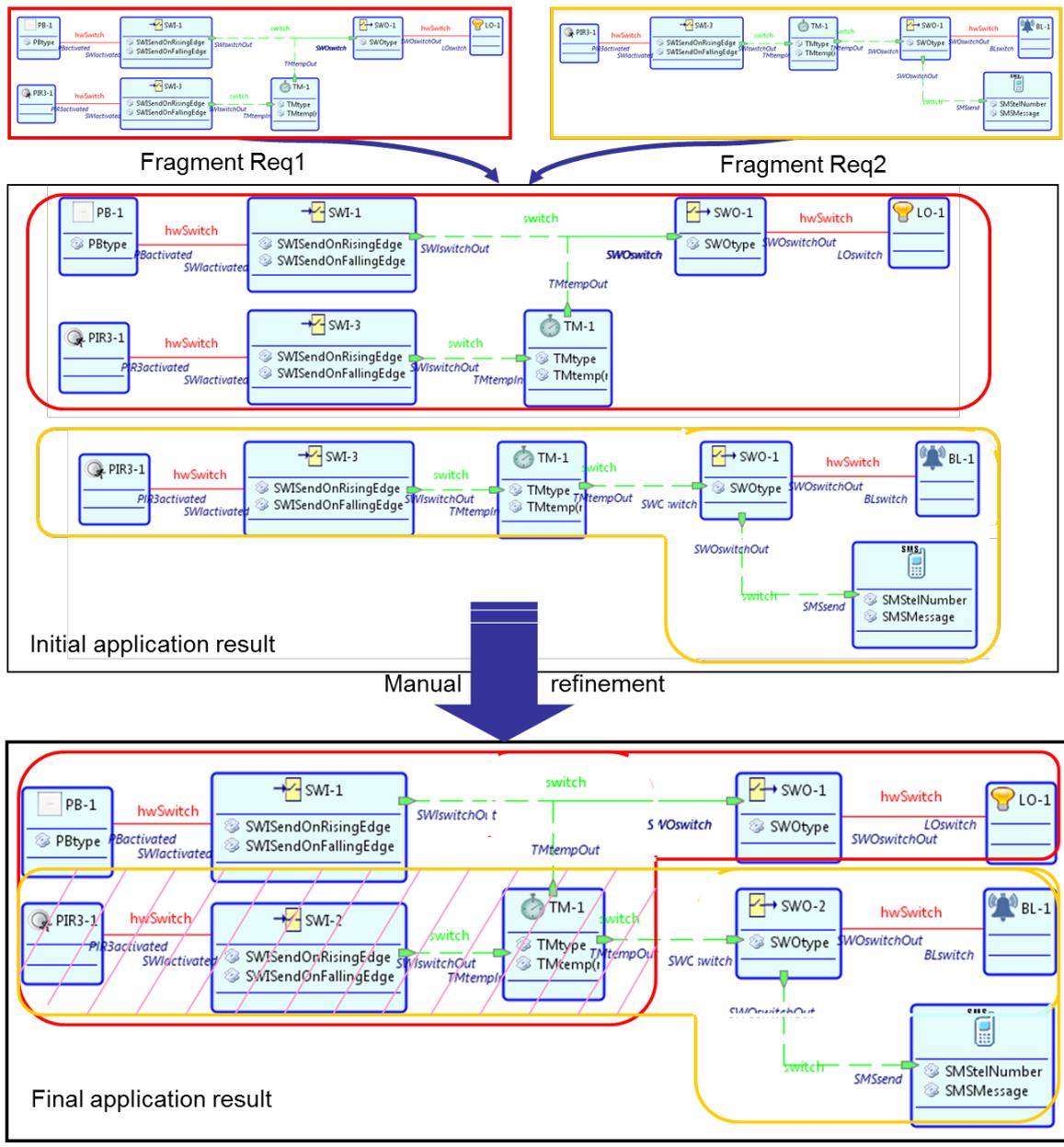


Figura 3. Ejemplo de reúso de fragmentos de modelos doméstico

## Referencias

- [1] S. Kelly and J.P. Tolvanen, "Domain-Specific Modeling: Enabling Full Code Generation", John Wiley & Sons, 2008.
- [2] C.W. Krueger, "Software reuse", ACM Computing Surveys, vol. 24(2), pp. 131-183, 1992.
- [3] M. Jimenez, F. Rosique, P. Sánchez, B. Álvarez, A. Iborra, Habitation: A Domain-Specific Language for Home Automation, IEEE Software, vol. 26(4), pp. 33-38.
- [4] Fernando Molina and A. Toval, "Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems", Advances in Engineering Software, Elsevier, vol. 40, pp. 1306-1317, 2009.
- [5] A. Toval, B. Moros, J. Nicolás, and J. Lasheras, "Eight key issues for an effective reuse-based requirements process", International Journal of Computer Systems Science and Engineering 23 (6) (2008) 1–20.
- [6] C. Vicente-Chicote, B. Moros and A. Toval, "Remm-studio: an integrated model-driven environment for requirements specification, validation and formatting", Journal of Object Technology, vol. 6 (9), pp.437–454, 2007.
- [7] IEEE Std 830-1998, "IEEE Recommended Practice for Software Requirements Specifications", IEEE Computer Society, 1998.
- [8] A. von Kethen, B. Paech, F. Kiedaisch, F. Houdek, "Systematic Requirements Recycling Through Abstraction and Traceability", 10th Intl. Requirements Engineering Conf, (RE'02), pp. 273- 281, 2002, Essen, Germany.
- [9] W. Lam, "A case-study of requirements reuse through product families", Annals of Software Engineering, 5, pp. 253-277, 1998.