

sistemática y reiterada los modelos a lo largo de todo el ciclo de vida de desarrollo del software. El uso de modelos permite aumentar el nivel de abstracción con que se realizan los diseños así como el nivel de reutilización de los mismos. Además, la utilización del enfoque MDE facilita la comunicación de ideas ya que éstas se pueden expresar de manera explícita (por lo general utilizando una notación gráfica asociada a los conceptos que se modelan) y no diluidas entre interminables líneas de código.

Dado que MDD solamente aporta un enfoque se hace necesario el uso de una arquitectura que implemente dicho enfoque. La *Arquitectura Dirigida por Modelos* (MDA) [44] es una de las propuestas de mayor interés en el desarrollo de software dirigido por modelos. MDA es la propuesta promovida por el consorcio Object Management Group (OMG). MDA propone que los sistemas software sean generados directamente a partir de modelos, la idea clave que subyace a MDA es que si el desarrollo está guiado por los modelos del software, se obtendrán beneficios importantes en aspectos fundamentales como son la productividad, la portabilidad, la interoperabilidad y el mantenimiento.

Para este fin, el marco de trabajo MDA especifica tres niveles de abstracción que proporciona tres puntos de vista diferentes, estos niveles son:

- Nivel **CIM**, nivel de modelos independientes de computación. Es el nivel de mayor abstracción y se centra en el dominio del sistema así como en sus requisitos específicos.
- Nivel **PIM**, nivel de modelos independientes de la plataforma. Este nivel de abstracción intermedio proporciona una descripción de la funcionalidad del sistema de forma independiente de la plataforma de implementación.
- Nivel **PSM**, nivel de modelos específicos de la plataforma. Es el nivel de menor abstracción, en el se define el sistema en términos de una plataforma específica.

De esta manera la división en distintos niveles de abstracción permite abordar el desarrollo de software desde distintos puntos de vista independientes, minimizando la complejidad del problema inicial.

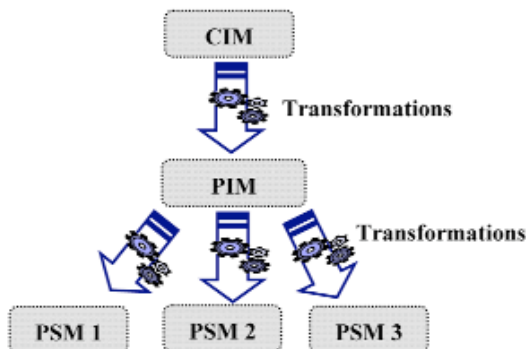


Figura 1: Niveles de abstracción MDA.

Existe una estrecha relación entre los diferentes modelos construidos en MDA. Los modelos más abstractos son la base para la construcción de los modelos específicos así como los modelos específicos son los que soportan los modelos de un nivel de abstracción menor (ver Figura1). Esta relación es representada en esta arquitectura por las transformaciones entre modelos, las cuales constituyen una de las características fundamentales de este enfoque.

Las **transformaciones** de modelos permiten pasar de un modelo origen a otro destino, de forma horizontal si pertenecen a un mismo nivel de abstracción o de forma vertical si son de distintos niveles (ver Figura1) de modo que se puede obtener un modelo PIM a partir de un modelo CIM, por ejemplo. Por último MDA permite obtener de forma automática o casi automática el código de la implementación del sistema con el uso de transformaciones modelo a texto.

Gracias a este enfoque de modelado dividido en niveles de abstracción se consigue dividir la complejidad, de modo que el usuario final puede trabajar a un alto nivel de abstracción de forma independiente al resto del desarrollo.

La metodología propuesta satisface tanto requisitos propios del enfoque MDE-MDA como las necesidades propias del dominio domótico, de modo que cumple los siguientes requisitos:

- Permitir especificar un sistema domótico utilizando **primitivas conceptuales adecuadas** que permitan capturar los requisitos a un nivel de gran abstracción, disminuyendo la complejidad del sistema.
- **Posibilidad de diseñar nuevos dispositivos**, de forma que no sea necesario seleccionar la infraestructura en las primeras etapas de desarrollo.
- Permitir **reutilizar** los elementos proporcionados en un **catálogo** de elementos preexistentes.
- Soporte completo al ciclo de vida de desarrollo, desde la captura de requisitos hasta la obtención de la implementación del sistema.
- Aumentar la calidad del software y la productividad.

Con este planteamiento se ha adoptado la metodología que viene representada en la Figura 2.

Esta metodología comienza con la captura de requisitos a un alto nivel de abstracción (parte superior figura 2). Para ello el nivel de origen es el **CIM** (Modelo Independiente de Computación) con el que se modelan los requisitos del sistema, describiendo la situación en que será usado el sistema, sirviendo tanto como ayuda para entender el problema como una base de vocabulario para usar en los demás modelos.

La captura de los requisitos iniciales del sistema se produce a través de la especificación de **lenguajes de modelado** cuya sintaxis abstracta es definida por **metamodelos**. En esta metodología resulta más interesante utilizar *Lenguajes Específicos de Dominio* (DSL) particularizados para el dominio domótico que permita formalizar modelos con mayor grado de expresividad proporcionando unas primitivas más próximas al dominio del problema. Este DSL (parte superior izquierda de la figura 2), permite integrar en el modelado tanto elementos pre-existentes en el mercado (recogidos en un catálogo de elementos domóticos) como nuevos elementos.

En un nivel de abstracción intermedio (**PIM: Modelo Independiente de la Plataforma**) se define la funcionalidad del sistema manteniendo los requisitos de forma independiente de la plataforma. Para llevar a cabo esta tarea se utilizará V3Studio [5], una herramienta para el diseño de sistemas basados en componentes siguiendo un enfoque dirigido por modelos que permite especificar la funcionalidad del sistema mediante el modelado de componentes, máquinas de estado y diagramas de actividad

Por último, en el nivel de menor abstracción, el nivel **PSM (Modelo específico de la Plataforma)** se seleccionará la plataforma/as de implementación y se obtendrá el código final ejecutable. De esta forma se obtendría por ejemplo para EIB/KNX [6] la implementación del sistema de forma automática o casi automática, pudiéndose retocar con la herramienta propia de EIB/KNX.

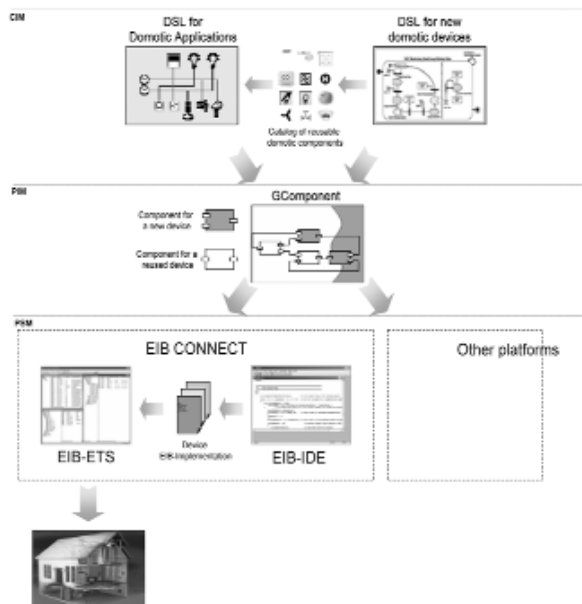


Figura 2: Metodología propuesta.

3. Conclusiones

En este artículo se ha presentado una metodología basada en el enfoque dirigido por modelos como alternativa para resolver los problemas encontrados en el desarrollo tradicional de software para sistemas domóticos.

Las implementaciones se adaptan a los cambios y facilitan su integración con otros sistemas. Además, se proporciona la abstracción necesaria para representar sistemas de una forma mucho más sencilla e intuitiva gracias al uso de lenguajes específicos del dominio domótico, por lo que el nivel de especialización requerido por el usuario no resulta tan elevado.

Por otro lado se favorece la validación en fases tempranas del desarrollo, ahorrando recursos tanto temporales como materiales y evitando errores en la implementación final.

Gracias a la incorporación de un catálogo de funcionalidades pre-existentes se aumenta la reutilización y la flexibilidad. La interoperabilidad deja de ser un problema gracias a la posibilidad de modelar nuevos dispositivos en un nivel de abstracción CIM de forma totalmente independiente a la tecnología de implantación final.

Como el proceso de desarrollo es independiente de las tecnologías utilizadas para la implementación del sistema final, se puede facilitar la validación en las primeras etapas del desarrollo.

En definitiva esta metodología contribuye a la evolución de los sistemas domóticos simplificando notablemente el proceso de desarrollo, proporcionando primitivas intuitivas, aumentando el nivel de abstracción, reduciendo la complejidad y aumenta la calidad y productividad del software obtenido.

4. Agradecimientos

Este trabajo ha sido financiado parcialmente por el gobierno español (CICYT, TIN2006-15175-C05-02) y el gobierno regional de Murcia (Programa Séneca).

Referencias

- [1] Ian Sommerville, "Ingeniería de Software.", 7ª Edición, Pearson, Addison Wesley, Madrid, 2005.
- [2] Carlos Fernández, Ignacio R. Matías, "El proyecto domótico. Metodología para la elaboración de proyectos y aplicaciones domóticas.". España, 2004.
- [1] Bran Selic, "The Pragmatics of Model-Driven Development". IEEE Software, 20(5):46-51, September/October 2003.
- [4] J. Miller, J. Mukerji, "MDA Guide Version 1.0.1.", OMG, 2003.
- [5] Diego Alonso, Cristina Vicente-Chicote, "V3Studio: Un entorno gráfico para el diseño de sistemas basados en componentes siguiendo un enfoque dirigido por modelos". JISBD 2007.
- [6] KNX standard disponible en <http://www.knx.org/knx-standard/introduction/>