

# Hacia el Modelado Conceptual de Sistemas Domóticos

Javier Muñoz, Joan Fons, Vicente Pelechano, Oscar Pastor

Depto. de Sistemas Informáticos y Computación  
Universidad Politécnica de Valencia  
Camino de Vera s/n, 46022  
Valencia

{jmunoz, jjfons, pele, opastor}@dsic.upv.es

**Abstract.** La introducción de las nuevas tecnologías en el hogar está dando paso a un nuevo tipo de sistemas de información: los sistemas domóticos. Actualmente el estado de las técnicas y métodos para el desarrollo de sistemas domóticos se encuentra poco evolucionado, utilizándose habitualmente tecnologías de bajo nivel de abstracción. El previsible crecimiento del área en un futuro cercano plantea la necesidad de contar con métodos robustos que permitan el desarrollo de estos sistemas en un nivel cercano al espacio del problema. En este artículo se presenta HAMON, un lenguaje para el modelado orientado a objetos de sistemas domóticos. Mediante la utilización del modelado orientado a objetos y la aplicación de los patrones de traducción oportunos, se pretende crear un método de generación automática de código que permita obtener sistemas domóticos de calidad.

## 1 Introducción

Aunque hace más de diez años que se trata sobre el área [1], es ahora cuando la domótica está empezando a penetrar en los hogares. Durante todo este tiempo la domótica se ha visto relegada a historias sobre el futuro, aunque ya existieran en el mercado algunas soluciones llamadas domóticas. Gran parte de la responsabilidad de este estancamiento ha recaído sobre la falta de consenso sobre el significado del propio concepto, aunque existan definiciones como la de [9]. Ante esta carencia han aparecido interpretaciones interesadas desde algunos sectores que han impedido la existencia de una visión clara del área que permitiera una evolución bien definida.

En el resto de esta introducción se ofrecerá una visión de la domótica, las causas de su actual resurgimiento y el impacto futuro que esta área va a tener en la sociedad. En el segundo apartado se justificará la necesidad de disponer de métodos robustos para el desarrollo de sistemas domóticos, analizando las tecnologías actuales y comparando su evolución con otras áreas. En el tercer apartado se mostrará la aproximación adoptada en esta propuesta, describiendo las primitivas de un lenguaje para el modelado conceptual de sistemas domóticos. Finalmente, en el cuarto apartado se han incluido las conclusiones de este artículo, trabajos relacionados con este y los posibles trabajos futuros.

## 1.1 Una Definición de Sistema Domótico

Etimológicamente el término domótica se refiere a la “automatización del hogar”. En inglés este concepto se conoce como “*home automation*”. Cuando el ámbito de la automatización aumenta hasta abarcar un edificio o inmueble pasa a denominarse inmótica (“*building automation*” en inglés). También se maneja la expresión “*computer automated building*” para destacar la existencia de ordenadores encargados de realizar el control del edificio.

Por otra parte, la gran difusión de los ordenadores e Internet en los hogares está originando nuevas necesidades en los sistemas domóticos [8]. Existe actualmente un debate en la comunidad de desarrolladores e instaladores de soluciones domóticas a este respecto. Por una parte se encuentran los que opinan que la domótica debe únicamente abordar la automatización del hogar. Por otra parte los hay que opinan que la funcionalidad para áreas como las redes de datos de ámbito residencial y la conectividad con redes externas también debe ser cubierta. En ocasiones se maneja el término “hogar digital”, como evolución de domótica, para referirse al segundo concepto.

En este artículo se considerará que **un sistema domótico es aquel sistema informático encargado de proporcionar servicios en el ámbito del hogar o, en general, los edificios**. Esta definición es muy abierta, por lo que en los siguientes párrafos se pasará a describir con mayor detalle.

Como sistema informático, un sistema domótico se compondrá de software y hardware. Dentro del hardware se incluyen los elementos que permiten recibir información del entorno y los elementos que permiten efectuar acciones sobre el entorno. Además, también se contempla la existencia de redes entre las distintas unidades que componen el sistema.

La principal caracterización de la domótica se realiza a través de los servicios que ofrece [8]. Estos servicios tienen un denominador en común: el ámbito en el que se ofrecen. Se considerará que la domótica ofrece servicios en las siguientes cuatro áreas:

- **Automatización y control.** En esta área se engloba la funcionalidad que se ha relacionado tradicionalmente con la domótica. Se trata de servicios de automatización y control principalmente de dispositivos domésticos (lavadora, nevera, etc.) y elementos de la casa (luces, toldos, etc.). También se incluyen servicios de control energético (calefacción, ventilación, etc.)
- **Seguridad y vigilancia.** Seguridad anti-intrusos, seguridad ante accidentes y servicios de salud.
- **Comunicaciones.** Servicios de intercomunicación entre las distintas redes del hogar, comunicación con redes externas, compartición de dispositivos, etc.
- **Tele servicios y ocio.** Servicios de video-bajo-demanda, juegos en red, telebanca, control remoto de los dispositivos del hogar, etc.

## 1.2 Estado de Evolución de la Domótica

La domótica es todavía un área joven, aunque algunas de las tecnologías que utiliza son más maduras. Hace unos quince años gozó de gran popularidad y difusión en los medios de comunicación, pero la baja calidad de los sistemas, la carencia de una divulgación rigurosa y su alto precio frenaron su penetración en la sociedad y su evolución.

Tras una época de transición, la domótica se encuentra ahora con mejores condiciones: la evolución de las tecnologías necesarias, la aparición de estándares, el interés de promotores y constructores, y la penetración de Internet están propiciando su difusión.

Una muestra de este renacimiento es el creciente número de productos y sistemas disponibles en las distintas áreas nombradas anteriormente. Por tanto, hay numerosos factores que hacen prever la inminente eclosión del área con el consiguiente aumento tanto del número de sistemas domóticos existentes como de la complejidad de los mismos.

## 2 La Necesidad de Métodos Robustos

Actualmente gran parte de la programación de la funcionalidad del sistema se realiza a muy bajo nivel. En su mayor parte se trata de sistemas centralizados donde todos los **dispositivos se encuentran cableados a una unidad central**. Los micro-controladores que gestionan el sistema se programan en lenguaje ensamblador. Esto proporciona un fuerte grado de estabilidad y robustez, pero complica enormemente la modificación o ampliación del sistema.

Otra posible opción es la utilización **de redes de control**. Esta posibilidad presenta el inconveniente de la falta de un estándar ampliamente reconocido que permita la interoperabilidad entre un gran número de dispositivos. Si bien es cierto que existen numerosos protocolos abiertos y estándares oficiales [2][3], cada uno con sus ventajas e inconvenientes, no hay un estándar *de facto*. Por tanto, aunque se opte por utilizar redes de control, el sistema final también se verá limitado por la oferta de dispositivos que soporten el protocolo utilizado.

Ante la carencia de un estándar, son muchos los fabricantes que han optado por desarrollar **sistemas cerrados**. De este modo se posibilita el control total del protocolo pudiendo ajustarlo para una necesidades concretas. El objetivo de estas iniciativas es conseguir una fuerte dependencia de los clientes, que tampoco encuentran una alternativa evidente.

En estos dos últimos casos la programación de las redes se realiza mediante lenguajes imperativos de bajo nivel utilizando las primitivas que ofrece el protocolo utilizado. En las clasificaciones de herramientas y *frameworks* propuestos en [4] y [9] se puede observar que estas se encuentran fuertemente ligadas a un tecnología en concreto, como en [7].

La evolución de las técnicas de desarrollo de sistemas domóticos está siendo similar a la evolución en otras áreas de la informática. Tanto en los sistemas de gestión empresarial como posteriormente en el desarrollo web, inicialmente se

utilizaban tecnologías muy cercanas al espacio de la solución. Cuando la complejidad de estos sistemas aumentó surgió una serie de graves problemas (entregas fuera de plazo, descontrol de costes, sistemas de mala calidad, etc.) que desembocaron en una crisis. Para abordar estos inconvenientes se han desarrollado una serie de métodos y tecnologías que acercan el desarrollo de estos sistemas al espacio del problema.

En la domótica nos encontramos todavía en una fase inicial, donde los sistemas no son excesivamente complejos. Pero, como se vio en el punto anterior, es previsible un inminente aumento en el número y complejidad de los mismos que puede desencadenar una crisis en el sector. El objetivo del trabajo realizado en esta línea de investigación es aprovechar esta oportunidad para evitar una previsible “crisis de las aplicaciones domóticas”. Para ello se propone la utilización del modelado conceptual orientado a objetos en esta área de manera análoga a como se ha realizado en otros dominios.

### 3 Modelado Conceptual de Sistemas Domóticos

La utilización del modelado conceptual para el desarrollo de sistemas domóticos tiene una serie de ventajas. Mediante esta aproximación el proceso de desarrollo es más intuitivo, ya que se utilizan primitivas más cercanas al espacio del problema. Además, utilizando los lenguajes de modelado adecuadamente se puede acelerar el proceso de producción, debido a que es posible la generación del software para la gestión del sistema directamente a partir del modelo. De este modo el proceso de desarrollo es independiente de las tecnologías utilizadas para la implementación del sistema final, a la vez que puede facilitar una validación temprana de los requisitos mediante la generación de prototipos o la animación de los modelos.

El trabajo en esta línea se ha iniciado con un estudio de los sistemas domóticos actuales y las tecnologías utilizadas para su desarrollo. Con esta información se ha realizado un análisis de las necesidades expresivas para el modelado de sistemas domóticos. Partiendo de estas se está construyendo el lenguaje que se describe en el siguiente apartado.

#### 3.1 HAMON (Home Automation Modeling Language)

En este apartado se describe HAMON (Home Automation Modeling Language), el lenguaje propuesto para el modelado de aplicaciones domóticas. Ha sido creado partiendo de las siguientes premisas:

- Debe facilitar la **máxima expresividad posible a partir de un número mínimo de primitivas**. Además, los modelos creados deben tener el nivel de detalle necesario para la posterior generación automática del software del sistema.
- En segundo lugar **debe evitar ambigüedades** que podrían causar confusión en la interpretación de los modelos y, además, impediría la generación de código de un modo riguroso y completo.

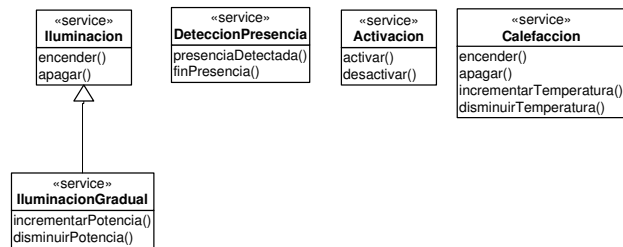
- [illegible]

Para ilustrar la descripción del lenguaje se va a utilizar un pequeño sistema de ejemplo. Se pretende que cuando el usuario encienda la luz del salón se active también la regulación de temperatura. Por otra parte si alguien entra en la sala se encenderá la luz automáticamente, pero no se iniciará la regulación de temperatura, ya que se considera que el usuario ha entrado en la sala transitoriamente. Debe existir también iluminación para lectura activable de manera independiente a la iluminación general. En la figura 1 se muestra un posible plano del salón.

- Un **servicio** del hogar, o *HomeService*, es una agrupación funcional semánticamente relacionada. Se compone de un conjunto de operaciones. Para describir el comportamiento de los servicios se les asociará un diagrama de transición de estados (DTE).
- Una **relación de agregación** expresará una agrupación de servicios. Como caso específico, la **composición** de servicios significará una agregación fuerte entre servicios donde el servicio componente no puede existir independientemente del servicio compuesto. Será necesario

proporcionar una semántica bien definida a esta relación utilizando un marco como el propuesto en [10].

- Una **relación de generalización**, tal y como se define en [5], es una relación taxonómica entre un elemento más general y uno más específico. Análogamente a las relaciones de composición, será necesario proporcionarles de una semántica bien definida utilizando un marco como el propuesto en [11].

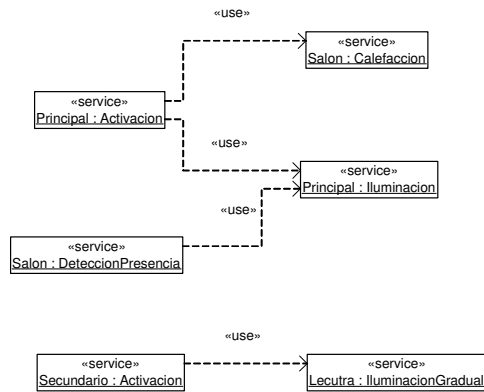


**Fig 2.** Catálogo de servicios del sistema de ejemplo. Está compuesto de cinco tipos de servicios, dos de los cuales se encuentra relacionados taxonómicamente.

El catálogo de servicios del sistema utilizado como ejemplo se compondrá de cinco tipos de servicios, tal y como se muestra en la figura 2. El servicio de *Activación* será el utilizado por el usuario manejar la iluminación. También se muestra una relación taxonómica entre el servicio de *Iluminación* y el servicio de *IluminacionGradual* que, además de la funcionalidad ofrecida por el servicio padre, permite modificar la potencia del alumbrado.

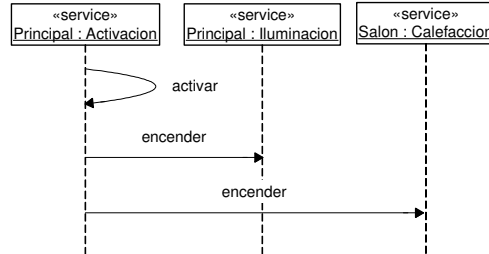
**Modelado del sistema.** El objetivo de la etapa de modelado es realizar una descripción global del sistema que sea independiente de la arquitectura física final del mismo. Para ello se identificarán los servicios necesarios para satisfacer las necesidades del sistema, así como sus interacciones. Se realizarán dos tareas para completar esta etapa:

1. En primer lugar se añadirán al diagrama **instancias de los servicios** necesarios para cumplir los requisitos del sistema. Es necesario realizar el modelo con objetos debido a que, aunque todas las instancias presenten un mismo comportamiento (descrito por su DTE), la funcionalidad podrá ser distinta. Existe un claro ejemplo de esto en el sistema de muestra, donde los dos servicios de activación realizan funciones distintas. También se establecerán **relaciones de dependencia de uso** entre los objetos para expresar que una instancia necesita usar operaciones de otra instancia.
2. Se describirán las interacciones entre las distintas instancias de los servicios. Para ello se asociarán **diagramas de interacción** (secuencia) de UML [5] al modelo del sistema. Los mensajes en estos diagramas deberán ser consistentes con las relaciones de uso expresadas en el diagrama de instancias de los servicios.



**Fig 3.** Modelo del sistema de ejemplo. Se muestran las instancias de los distintos servicios que conformarán el sistema final.

En la figura 3 pueden observarse las instancias de los distintos servicios. Existirá un servicio de *Iluminacion* para la iluminación principal del salón y un servicio de *IluminacionGradual* para la iluminación de lectura. También serán necesarios dos servicios de *Activacion*, uno exclusivo para la luz de lectura y otro para la activación de la iluminación principal y la calefacción. Finalmente, existirá una instancia del servicio *DeteccionPresencia* para detectar la entrada de usuarios en el salón.



**Fig 4.** Diagrama de interacción entre la instancia del servicio de activación y las instancias del servicio de iluminación general y del servicio de calefacción.

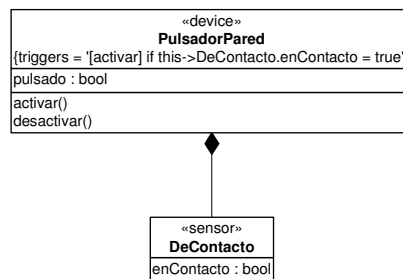
En el diagrama de interacción mostrado en la figura 4 puede verse cual es la secuencia de mensajes que debe ocurrir al ser invocada la operación *activar* de la instancia *Principal* del servicio de *Activacion*. De este modo se refleja la necesidad de que se enciendan simultáneamente la iluminación principal y la calefacción.

**Especificación detallada del sistema.** El objetivo de la especificación detallada del sistema es enriquecer el modelo con detalles de la arquitectura y la funcionalidad interna de cada instancia de un servicio. Para ello se asociará a estas un diagrama de clases formado por elementos de las siguientes primitivas:

- Un **dispositivo**, que representa un dispositivo de la realidad. Se compone de un conjunto de atributos, operaciones, disparos, sensores y efectores. Los disparos se especificarán utilizando un lenguaje de *triggers*.
- Un **sensor**, que representa un componente físico que adquiere datos del entorno del sistema. Se compondrá únicamente de un conjunto de atributos que sólo podrán ser consultados, ya que su modificación se realizará de manera externa al sistema.
- Un **efector**, que representa un componente físico que realiza acciones sobre el entorno del sistema. Se compondrá únicamente de un conjunto de operaciones.
- Existirán **relaciones de composición**, donde el elemento compuesto siempre será un dispositivo y los componentes podrán ser sensores o efectores.

Cada uno de estos diagramas contendrá al menos un dispositivo. Toda operación del servicio al que se encuentra asociado el diagrama deberá aparecer en uno, y solo uno, de los dispositivos. Un mismo dispositivo puede aparecer los diagramas asociados a varias instancias de servicios.

La descripción de la funcionalidad interna de cada servicio se realizará mediante la descripción de las operaciones de los dispositivos. Cada operación tendrá asociada una serie de acciones descritas en un lenguaje de acciones semánticas (ASL) utilizado de forma disciplinada para garantizar que se cumple el comportamiento especificado en los diagramas de interacción del modelo.



**Fig 5.** Diagrama de clases asociado a la instancia *Principal* del servicio *Activador*.

Del diagrama mostrado en la figura 5 se desprende que la instancia *Principal* del servicio *Activador* será realizada por un *PulsadorPared*. Este dispositivo se compone de un sensor que detecta cuando el usuario ha apretado el pulsador

Finalmente, en la figura 6 se muestran las acciones en un lenguaje de acciones semántica para la activación del pulsador. En dicho código aparecen las llamadas para encender el servicio de iluminación principal y el servicio de calefacción, tal y como se describía en el diagrama de interacción asociado a dicha operación.

```

this.pulsado = true

this->Principal.encender()

this->Salon.encender()

```

**Fig 6.** Código en un ASL asociado a la operación *activar* de la instancia *Principal* del servicio de *Activacion*.

Una vez realizado este último paso el sistema se halla completamente especificado. Como puede observarse, el modelado del sistema es independiente de la especificación detallada. Por tanto un mismo modelo podría ser realizado por varias especificaciones detalladas.

Por otra parte es importante hacer notar que la especificación detallada de cada instancia de un servicio no tiene por qué realizarse desde cero. Es posible asociar a cada servicio del catálogo una o varias especificaciones detalladas para sus instancias. Partiendo de estas únicamente sería necesario añadir las acciones para realizar las invocaciones descritas en los diagramas de interacción del modelo del sistema.

## 4 Conclusiones

En este artículo se ha realizado una propuesta de un lenguaje de modelado orientado a objetos para la descripción de sistemas domóticos. Con este lenguaje se pretende contribuir a la evolución de las técnicas de desarrollo de este tipo de sistemas acercando el proceso de desarrollo al espacio del problema.

[6] también tenía como objetivo aumentar el nivel de abstracción del desarrollo, pero se encuentra limitado a una única tecnología. Por ello, las primitivas utilizadas son el reflejo directo de las primitivas de la tecnología objetivo. Por otra parte, los esquemas no tienen una representación gráfica, sino que se deben describir mediante LISP.

HomeNet Tool es una herramienta gráfica presentada en [4] que permite la programación de la funcionalidad del sistema relacionando eventos de los diversos dispositivos. Aunque es una aproximación bastante cercana a la propuesta en este artículo, carece de toda la potencia que proporciona el modelo completo orientado a objetos.

A pesar de que VISIR [9] se presenta como una herramienta para la simulación de un sistema domótico, podría considerarse desde el punto de vista del desarrollo. También es completamente independiente de tecnologías, aunque presenta una aproximación muy ligada a los dispositivos físicos del sistema. Además, dentro de su catálogo de dispositivos que pueden formar parte de un esquema no se consideran aquellos orientados al entretenimiento y las comunicaciones.

Todavía hay mucho trabajo que realizar en esta línea de investigación. Junto con el lenguaje propuesto es necesario contar con un método bien definido e intuitivo que guíe su utilización. Actualmente se está trabajando en la descripción de dicho método considerando las distintas aproximaciones posibles. Los pasos a seguir para la construcción de los modelos serán diferentes en el caso de proyectos completamente

configurables y en los casos en los que existan restricciones sobre los dispositivos que se deben utilizar, por ejemplo.

Paralelamente a la tarea anterior, también se está trabajando en la creación de los patrones de traducción entre las primitivas conceptuales descritas en el lenguaje y la tecnología de implementación utilizada. Estos patrones permitirán la generación automática del software de los sistemas domóticos a partir de los modelos creados.

Por otra parte, se prevén futuras ampliaciones del lenguaje para considerar necesidades expresivas adicionales. Las extensiones al núcleo del lenguaje presentado consistirán en primitivas conceptuales para expresar la existencia de objetos multimedia, características de distribución, soporte de sistemas legados, restricciones temporales, etc.

## Referencias

1. Ryan, J.L. : Home Automation. Electronics & Communication Engineering Journal , Volume: 1 Issue: 4 , July-Aug. 1989. Page(s): 185 -192
2. Wacks, K. P.: International Development of Home Automation Standards. Consumer Electronics, 1992. Digest of Technical Papers. ICCE., IEEE 1992 International Conference on , 2-4 June 1992. Page(s): 274 -275
3. Hwang, M., Jeon, Y., Kim, J.: Standarization Activities and Technology Competitors for the In-Home Networking. Internation Conference on Communication Technology, 1998.
4. Leeb, G., Posta, R., Schildt, G.H. : A Configuration Tool for Homenet. Consumer Electronics, IEEE Transactions on , Volume: 42 Issue: 3 , Aug. 1996. Page(s): 387 -394
5. Object Management Group: OMG Unified Modeling Language Specification. Versión 1.5, Marzo 2003
6. Eden, F.: Modelling CEBus Home Automation with Knowledge Based Tools. IEEE Transactions on Consumer Electronics, Vol. 36 N° 4, Agosto 1990.
7. Bertsch, L.A.: Development Tools for Home Automation. IEEE Transactions on Consumer Electronics, Vol. 36, N° 4, Noviembre 1990.
8. Chen, W. Y. : Emerging Home Digital Networking Needs. Community Networking Proceedings, 1997 Fourth International Workshop on , 11-12 Sept. 1997. Page(s): 7 -12
9. González, V. M. et al.: VISIR, A Simulation Software for Domotics Installations to Improve Laboratory Training. 31<sup>st</sup> ASEE/IEEE Frontiers in Education Conference, 2001.
10. Albert, M., Pelechano, V., Fons, J., Ruiz, M., Pastor, O.: Implementing UML Association, Aggregation and Composition. A Particular Interpretation based on a Multidimensional Framework. 15th Conference On Advanced Information Systems Engineering (CAISE2003). Pendiente de publicación. 2003.
11. Pelechano, V.: Managing Taxonomic Relationships in Automatic Software Production Enviroments: A Pattern based Approach. UMI ProQuest Digital Dissertationns. 2001.