

## Aplicación del razonamiento cualitativo al “Hogar Digital”

Juan Antonio Álvarez<sup>1</sup>, Juan Antonio Ortega<sup>1</sup>, Jesús Torres<sup>1</sup>, Alejandro Fernández<sup>1</sup>, Manuel David Cruz<sup>1</sup>, Cecilio Angulo<sup>2</sup> y Francisco Velasco<sup>3</sup>.

<sup>1</sup> Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla.  
Avda. Reina Mercedes s/n. Sevilla.

{juan,ortega,jtorres}@lsi.us.es, {alejandro.fdez, manuel3086}@gmail.com

<sup>2</sup> Ingeniería de Sistemas, Automática i Informàtica Industrial, Escola Politècnica Superior d'Enginyeria de Vilanova i la Geltrú.  
cecilio.angulo@upc.edu

<sup>3</sup> Departamento de Economía Aplicada I, Universidad de Sevilla. Avda. Ramón y Cajal, s/n  
velasco@us.es

### Resumen

Identificar y reproducir el razonamiento humano, es uno de los objetivos principales de la Inteligencia Artificial. En el ámbito del hogar digital dicho objetivo cobra una gran importancia, pero aún no se ha logrado completar.

En este artículo comentamos la arquitectura propuesta para la realización de una vivienda domótica basada en la especificación realizada por OSGi. En ella, resolvemos la interacción con el usuario para que éste, a través de interfaces Web, pueda controlar su residencia. Dichos interfaces deben ser amigables con los habitantes del hogar digital y dará respuestas en su mismo lenguaje.

El control que le permitimos es completo, accediendo a todo tipo de dispositivos electrónicos a través de una gran diversidad de redes y protocolos. Además todas las acciones realizadas sobre los elementos de la casa, se registran en una base de datos. Ante esta potencia de actuación y recopilación de datos inmediata, creemos que hemos de dar un paso hacia la inteligencia de la vivienda a través del razonamiento cualitativo y concretamente usando lógica difusa sobre los datos obtenidos. De ese modo conseguiremos realizar una interacción inteligente y adaptable del hogar con sus habitantes.

años. Con la aparición de “Open Services Gateway Initiative” (OSGi) [10] en 1999, sectores de construcción, empresas de software y compañías de telecomunicaciones, colaboran para obtener productos aplicables tanto al hogar digital como a cualquier entorno habitable, incluyendo incluso los automóviles. Dicha iniciativa, pretende crear pasarelas de servicios a través de una arquitectura abierta y flexible, suministrando dichos servicios desde redes de área extensa hacia LANs o redes de dispositivos.

La creciente demanda de servicios de gestión y control de nuestro hogar por parte de la sociedad de la información, hizo que surgiese el proyecto Domoweb, cuyo objetivo principal, es la creación de un entorno domótico gestionado vía Web. El desarrollo de dicho trabajo nos ha permitido acumular experiencia sobre las arquitecturas orientadas a servicio, concretamente sobre la plataforma basada en la especificación de OSGi e implementada con software de fuentes abiertas. En este artículo, comentamos la estructura y servicios que oferta el proyecto y ampliamos su visión inicial, hacia ámbitos de aplicación de lógica difusa para conseguir interaccionar de manera inteligente con los habitantes de la vivienda. Describiremos interesantes escenarios donde podremos comprobar la mejora del proyecto inicial, en el que hemos comenzado a aplicar lógica difusa en diversos aspectos como interacción con los usuarios y predicción de actuaciones. Además, dejaremos temas abiertos para permitir la comunicación con grupos investigadores de razonamiento cualitativo y diagnosis.

### 1 Introducción

El problema de la extracción de conocimiento a partir de valores cualitativos ha sido resuelto en diferentes ámbitos. Desde la economía [12] hasta la educación [1] pasando por muchos otros contextos.

En este trabajo vamos a buscar un nuevo ámbito de aplicación: “El Hogar Digital”. La revolución para conseguir un entorno preparado para computación y comunicación ubicua, junto con una interacción a través de interfaces inteligentes, ha avanzado a pasos agigantados en los últimos

Este artículo está organizado del siguiente modo: En la sección 2 desarrollaremos la visión del proyecto Domoweb y mostraremos aspectos de su diseño. En la sección 3 describiremos el trabajo relacionado con la implementación que estamos realizando mediante lógica difusa para conseguir razonamiento cualitativo. Tras justificar el porqué de usar esta lógica, mostraremos los escenarios con los que quisimos ampliar la idea inicial. Finalmente en el cuarto apartado explicaremos las conclusiones obtenidas y apuntaremos el trabajo futuro.

## 2 Desarrollo

A continuación estudiamos la visión inicial del proyecto Domoweb y los servicios que ofrece.

### 2.1 El proyecto inicial

El objetivo primordial de Domoweb, es el de diseñar y desarrollar un sistema domótico virtual, controlado vía Web y basado en fuentes abiertas. Con este nombre, definimos un sistema capaz de gestionar elementos típicos de la domótica como pueden ser sistemas de alarmas y sistemas de control del confort del usuario, mediante los sensores y actuadores necesarios para ello. Una de las particularidades de este sistema es su capacidad de ser gestionado virtualmente, a través de interfaces adaptativas por el usuario final de manera fiable, segura y fácil. De este modo además de poder realizar operaciones de control y supervisión desde dentro de la vivienda, se podrían realizar desde cualquier punto con acceso a Internet. Dichas operaciones suponen interactuar con dispositivos con diversos protocolos de comunicación como X-10 [15] ó tecnologías como USB [8] o Bluetooth [9].

Debido a que uno de los requisitos iniciales de nuestro proyecto era el de basarse en software de fuentes abiertas y que la especificación de OSGi es demasiado extensa para implementarla con los recursos humanos disponibles, optamos por utilizar implementaciones de software libre existentes. En concreto nos han sido de mucha utilidad OSCAR [5] y Knopflerfish [13], que siguen la especificación de OSGi R3[11].

OSCAR es muy estable, y es usada por multitud de desarrolladores y programadores en distintos proyectos. El framework está pensado para experimentación de software orientado a servicios en general. Puede ser fácilmente incluido en otros proyectos y usado como un mecanismo de extensión. Gracias a OSCAR, podemos realizar ensamblado dinámico de aplicaciones, añadiendo y extendiendo funcionalidades de una aplicación en tiempo de ejecución, descargando sobre él nuevos componentes y servicios. Además proporciona técnicas de aseguramiento de la calidad de las aplicaciones que funcionan bajo ese entorno dinámico y basado en componentes.

Knopflerfish es otro desarrollo en código libre del framework especificado en OSGi R3. Como en el caso de OSCAR, está orientado a componentes y facilita la experimentación de software orientado a componentes y servicios.

Ambos proyectos presentan una interfaz muy intuitiva desde la que se nos permite instalar, desinstalar, iniciar, parar y actualizar componentes, además proveen repositorios para componentes que pueden analizarse ya que se facilita el código fuente.

### 2.2 Servicios ofrecidos

OSCAR y Knopflerfish ofrecen diferentes servicios, tanto de infraestructura como de aplicación, que ofrecen una im-

plementación adecuada para su uso. Para Domoweb, hemos utilizado y adaptado los siguientes:

- Log Service: Este servicio provee la interfaz necesaria para que se puedan registrar mensajes en la pasarela por parte de los servicios que lo requieran.
- Service Tracker: Mediante este servicio se puede llevar a cabo un proceso de vigilancia de servicios. Esto es muy útil para prevenir fallos por problemas de dependencias entre componentes.
- Configuration Admin: Se encarga de configurar los bundles (componentes OSGi) que requieran una configuración cuando son desplegados. De esta manera un bundle puede configurar a otro a través de este servicio.
- HTTP Service: Ofrece un servidor Web en el que se pueden registrar tanto servlets como recursos, de manera que sean accesibles desde el exterior.

Además de estos servicios, fue necesario implementar los siguientes:

- DBCommunications: Servicio de comunicación con sistemas de bases de datos.

Dicho servicio, gestiona el almacenamiento de manera persistente de la información relacionada con dispositivos, usuarios y el espacio domotizado. De entre las soluciones de almacenamiento persistente, se eligió el uso de un Sistema de Gestor de Bases de Datos Relacional (SGBDR) debido a que es una tecnología muy extendida en entornos web y en arquitecturas cliente servidor. En su desarrollo se usó la API JDBC [7] que nos independizó del SGBD usado.

- User Admin Service: Servicio de infraestructura descrito en la especificación de OSGi, en el que se detalla un modelo de autorización basado en roles o categorías de usuarios.

Este servicio, lo adaptamos perfectamente a nuestros requerimientos. A grandes rasgos, el modelo sugiere que todo servicio que ofrezca la pasarela esté asociado a un grupo de usuarios (categoría) y el usuario que quiere hacer uso de dicho servicio deberá pertenecer (o incluso ser él mismo, si el rol fuese una persona) a éste. Este servicio es interesante para el análisis cualitativo ya que la pasarela almacena un detallado registro de todo lo que sucede en la misma gracias a éste (los usuarios que han accedido, los dispositivos que se han puesto en marcha a una determinada hora, los intentos de intrusión a través de la red, etc.).

- Servicios de infraestructura que permiten la gestión y administración de dispositivos.

Domoweb ha implementado servicios que se comunican con dispositivos de tecnologías X10, USB y Bluetooth.

- Servicio de gestión de escenas.

Muy relacionado con la gestión de los dispositivos y el User admin., se encuentra el servicio de gestión de escenas. Una escena es una combinación o sucesión de eventos que se desarrollan en un espacio domotizado y que determina el estado en el que quedan los dispositivos del sistema. Escenas típicas pueden ser por ejemplo: "llegada a casa" (cuando algún residente entra a la vivienda, se encienden las luces necesarias para la visibilidad, se suben las persianas, se reproducen los mensajes del contestador, etc.); "visualización de películas" (se apagarían las luces y se encendería la TV. etc.); "marcharse de vacaciones" (se activa la alarma y la simulación de presencia) etc. Gracias a este servicio, los usuarios pueden añadir configuraciones personalizadas para cubrir sus necesidades. En la figura 1 podemos ver el interfaz de dicho servicio.

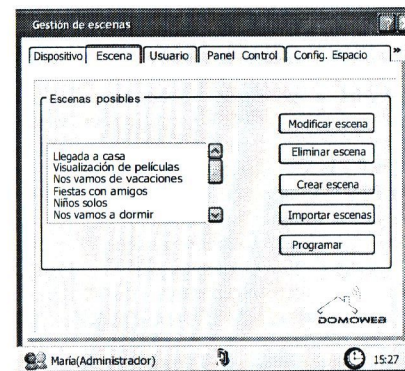


Figura 1. Interfaz para la gestión de escenas

En cuanto al método de autenticación, Domoweb ha optado por el mecanismo convencional de login y password, aunque la implementación queda abierta a cualquier otro sistema debido a la gran variedad que hay actualmente en el mercado. Dicho método no se ha especificado mediante un servicio de la pasarela, sino mediante un módulo Web en los interfaces de usuarios.

Debido al dinamismo que el entorno de ejecución que OSGi propone, consideramos muy importante determinar la manera y el orden en el que los distintos servicios deben desplegarse en la pasarela durante el inicio del sistema.

Al proceso de descarga e instalación de los elementos necesarios para que nuestro sistema pueda comenzar a funcionar correctamente, lo denominaremos aprovisionamiento inicial. Con tal fin, en Domoweb, hemos empleado el Oscar Bundle Repository (OBR)[5]. OBR es un servicio que permite la descarga e instalación de servicios de manera remota a través de un repositorio descrito en XML. Los distintos repositorios pueden ser accedidos de dos formas, a través de la web o mediante un acceso programado en la pasarela

haciendo uso de algún servicio que lo haga posible. Es responsabilidad del operador de la pasarela decidir qué repositorio de servicios será el empleado en cada caso e indicar si se tiene autorización para poder acceder a ellos y poder usarlos.

- StartupService: La necesidad de garantizar que el sistema arranca de manera adecuada cuando se inicia por primera vez, nos obliga a implementar un servicio de puesta en marcha.

Éste, descarga los componentes software básicos o de infraestructura que necesita el sistema para comenzar a funcionar. Para ello realiza las peticiones pertinentes a OBR y es éste último el que realiza la descarga e instalación de los componentes solicitados.

En el caso de nuestro estudio, debido a las funcionalidades que caracterizan cada servicio, se decidió desplegar los componentes en un determinado orden. En primer lugar se despliega el servicio de comunicación con las bases de datos. A continuación el servicio de registro de eventos en el sistema. Y por último los servicios de administración de usuarios y de control y gestión de dispositivos. Estos servicios básicos o de infraestructura son a su vez dependientes de otros servicios básicos de OSGi, lo que obliga a OBR a resolver estas dependencias.

## 3 Razonamiento cualitativo

Una vez visto todos los servicios que ofrece Domoweb, y tras comenzar la etapa de construcción, detectamos posibles mejoras para ofrecer más funcionalidad en la plataforma.

El elevado número de eventos: registro de servicios, actuaciones por parte de los usuarios, modificación de los dispositivos actuadores y valores medidos por los sensores, eran captados y almacenados por el servicio DBCommunication. La base de datos crecía con información que creímos que podía ser útil para poder dotar de capacidad de razonamiento a la vivienda. Ante esta situación decidimos aplicar técnicas de razonamiento cualitativo. Concretamente, creímos que la lógica Difusa o lógica Fuzzy podría ser bastante útil. Aunque comenzó a investigarse a mediados de los sesenta por Lotfy A. Zadeh [16] y encontró una fuerte resistencia entre la comunidad científica, actualmente está totalmente asentada en sistemas de control y es la base de "The Berkeley Initiative in Soft Computing" [2]. Dicha lógica es una forma de razonamiento que permite incorporar en los sistemas de automatización esquemas típicamente humanos, de tipo lingüístico cualitativo. Aunque estos esquemas son inherentemente menos precisos que los numéricos, en muchas ocasiones aportan información más útil para el razonamiento mental.

Una de las principales características de esta lógica es su capacidad para operar con conceptos vagos o ambiguos propios del razonamiento cualitativo, fundado sobre un soporte matemático que permite extraer conclusiones cuantitativas a

partir de un conjunto de observaciones (antecedentes) y reglas cualitativas (base de conocimiento). Estas características la dotaban de la herramienta adecuada para conseguir nuestro propósito. Otro aspecto que nos hizo decidimos por la lógica fuzzy en nuestro proyecto, fue la gran cantidad de aplicaciones exitosas que la usan en el hogar. Por señalar algunas, el ajuste del ciclo de lavado y enjuague a partir del número de platos en lavavajillas, el establecimiento del programa de cocción en hornos microondas, la sintonización del color y texturas en televisores o la regulación de la cantidad de jabón usado en lavadoras. La posibilidad de reutilizar las reglas de dichas aplicaciones nos resultó muy atractiva.

Por todas estas razones creímos que en el ámbito de la comunicación y la interacción del hogar con sus habitantes, así como en la predicción de situaciones a partir de antecedentes cuantitativos y cualitativos, podrían mejorar nuestra idea inicial.

### 3.1 Aplicación de razonamiento a Domoweb

La integración de lógica fuzzy a nuestro proyecto podía resultar difícil ya que no habíamos contemplado dicho requisito en la planificación ni en el análisis del proyecto. La especificación de OSGi propone una arquitectura como la que podemos ver en la ilustración de la figura 2.

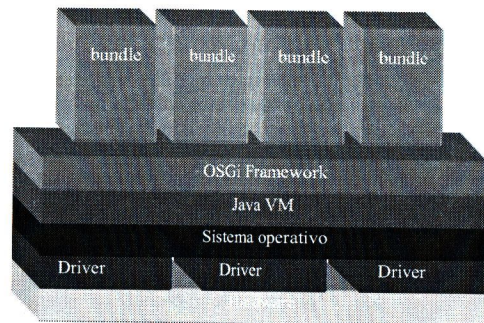


Figura 2. Arquitectura de una pasarela OSGi

En esta estructura, observamos como los servicios anteriormente descritos, están situados en la parte más alta de la figura, descritos como componentes o "bundles", que ofrecen dichos servicios. También podemos comprobar como la plataforma se apoya en la máquina virtual de Java y se independiza del hardware y el sistema operativo. Ante esta situación, buscamos una biblioteca para el manejo de la lógica difusa en Java. Encontramos FuzzyJ [4]. FuzzyJ es un conjunto de clases Java que pueden ser usadas para construir sistemas de lógica difusa. La API posibilita la representación y manipulación de información difusa. Actualmente estamos trabajando en su integración en diferentes bundles que ofrezcan el servicio de razonamiento difuso

El sistema propuesto por Domoweb se basa en la interacción con los usuarios. Responde a estímulos, llegados a través de sensores, producidos por actuaciones de éstos dentro del espacio domotizado. Por lo tanto la interacción con el usuario es esencial. Además, los servicios de inteligencia ambiental tratarían de ofrecer al usuario respuestas a los deseos predichos a partir de la base de conocimiento adquirida con anterioridad. Por tanto, necesitaremos de un componente software que cree, gestione, actualice y refine continuamente la base de conocimiento a través de un aprendizaje automático. El aprendizaje partiría de las observaciones que el sistema adquiriera a partir de las señales que recibe de los sensores y de los cambios de estado de los dispositivos a lo largo del tiempo, tratando de buscar patrones de comportamiento y hábitos de los usuarios.

Por otro lado el sistema debería albergar un segundo componente software que supervisara las cláusulas para disparar las reglas difusas. Un aspecto muy importante es que el sistema deberá ejecutar definitivamente la acción de una regla una vez se haya evaluado la probabilidad de acertar sobre el deseo de los usuarios. El umbral a partir del cual se considere segura la ejecución de una regla puede ser variable, dependiendo de como de crítica sea la acción a ejecutar.

Para mayor seguridad, el sistema podría auto-evaluarse durante un cierto periodo de tiempo. Para lograrlo, se simula el disparo de las reglas y la ejecución de las acciones y se observa cual es el comportamiento real de los usuarios. De esta manera podría conocer el porcentaje de acierto que hubiera obtenido en caso de haber disparado realmente las reglas de su base de conocimiento. Una vez pasado el periodo de tiempo de auto evaluación, el sistema podría establecer una base de conocimiento refinada sin que el usuario fuera afectado por un comportamiento no deseado del sistema durante el periodo de adaptación.

En el siguiente ejemplo, podemos observar como se podría llevar a cabo el proceso de aprendizaje, la creación de la base de conocimiento y el disparo de las reglas difusas.

A través de la observación del comportamiento de los usuarios, el sistema adquiere los siguientes patrones de comportamiento:

- Fernando enciende la lavadora antes de acostarse
- Lucía enciende el a/c cuando hace calor y se encuentra en la habitación.

El sistema traduce los valores del mundo real al ambiente fuzzy a través de su lógica. De esa manera se pueden traducir los valores reales de temperatura obtenidos por los sensores a valores difusos a través de funciones que establecen la probabilidad de encontrarse en un cierto rango. Para los dominios de Temperatura podríamos establecer las siguientes funciones características que se muestran en la figura 3:

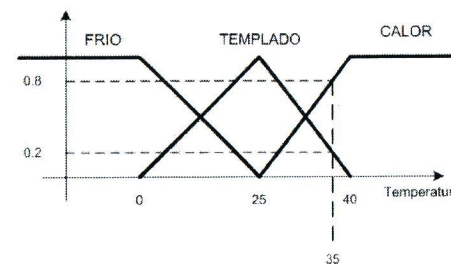


Figura 3. Funciones características para temperatura.

Como se observa, las funciones miembro traducen una temperatura = 35, en los valores difusos FRIO=0, TEMPLADO=0.2 y CALOR=0.8.

Consideremos ahora la regla: Si Temperatura=CALOR y EstaEn=HABITACION entonces A/C=INCREMENTAR. Supongamos que CALOR=0.8 y HABITACION=0.9, entonces la validez de la regla será de 0.8 (la del menor valor de los antecedentes) y la variable fuzzy INCREMENTAR sería también igual a 0.8.

Si otra regla actuara sobre la variable INCREMENTAR, está tomaría el menor resultado de ambos (consecuencias de realizar operaciones con esta lógica).

Una vez computadas las reglas fuzzy y evaluadas las variables, necesitaremos trasladar los valores nuevamente al mundo real. Proyectando el resultado anterior, sobre nuevas funciones de salida, buscaríamos un punto de equilibrio para obtener el valor más adecuado. Una manera de realizar esto es mediante el método del centro de gravedad, que en nuestra implementación supone que deberíamos incrementar la potencia del aire en un determinado porcentaje.

En FuzzyJ los conceptos difusos se representan con variables, conjuntos y valores difusos:

- Variable difusa: define los componentes básicos que son usados para describir un concepto difuso. Está constituido por su nombre, las unidades para medirla, el universo del discurso y un conjunto de términos difusos que describen el concepto difuso particular para esta variable.

Si quisiésemos definir la variable difusa 'temperatura' con los términos 'FRÍO', 'TEMPLADO' y 'CALIENTE', realizaríamos las siguientes instrucciones:

```
FuzzyVariable temp = new FuzzyVariable("Temperatura", -50, +50, "centígrados");
temp.addTerm("CALOR", {25,50}, {0,1},2);
temp.addTerm("FRIO", {-50,25}, {1,0},2);
temp.addTerm("TEMPLADO", "not FRIO and not CALOR");
```

- Conjunto difuso: es un mapeado de numeros reales a valores de referencia en el rango [0,1]. Por ejemplo:
 

```
double y[]={0, 0.5, 0.2, 1, 0.2, 0};
```

```
double x[]={0.2, 0.4, 0.8, 0.9, 0.95};
FuzzySet fSet=new FuzzySet(x, y, 5);
```

- Valor difuso: permite crear un concepto difuso para una determinada variable difusa a partir de una variable difusa previamente definida y un término previamente definido.

```
FuzzyValue fval = new FuzzyValue(temp, "very CALOR or FRIO");
```

Esta API también nos ayuda en la creación y validación de reglas difusas. A partir de las condiciones o antecedentes y los valores de entrada se evalúan las reglas que incluyamos en nuestro sistema. Un ejemplo de creación de reglas con FuzzyJ.

```
FuzzyRule calorHab = new FuzzyRule();
calorHab.addAntecedent(new FuzzyValue(temp,"CALOR"));
calorHab.addAntecedent(new FuzzyValue(estaEn,"HABITACION"));
calorHab.addConclusion(new FuzzyValue(a/c,"INCREMENTAR"));
```

### 2.5 Escenarios propuestos

Observando la domótica desde el punto de vista de la Inteligencia Ambiental, nos podríamos centrar en el aprendizaje automático. Si conseguimos integrar en nuestra arquitectura dicho aprendizaje, conseguiríamos que nuestro "Hogar Digital" se adapte a las necesidades de sus inquilinos, originadas a partir de las vivencias obtenidas día a día dentro de un entorno digital. Esto no es algo sencillo ya que los gustos y costumbres de las personas pueden diferir mucho unas de otras y además tienen la dificultad añadida de que el aprendizaje debe hacerse de una forma no intrusiva.

Posibles escenarios podrían ser los siguientes:

- Intuición de futuras acciones: Ante la elevada cantidad de observaciones almacenada en la base de datos del sistema, mediante razonamiento cualitativo, podríamos intentar prever las acciones más probables de los usuarios. Detectar cuándo hace "demasiado" frío o cuándo no hay "suficiente" luz, podría ser una situación en la que hogar se adaptase activando el calefactor o iluminando las zonas más lúgubres en las que se encuentra una persona cuyas "sensaciones" hubiesen sido recogido mediante la experiencia.
- Interacción inteligente: Las interfaces que provee Domoweb, no permiten una interacción inteligente. Mediante la lógica difusa podríamos hacer que nuestros interfaces diesen respuestas como: "Eres demasiado joven para activar el horno". Los conceptos de "demasiado" o "joven" se podrían definir como conjuntos nebulosos o fuzzy sets. Además podemos construir

subconjuntos, de manera que cualquier elemento pueda pertenecer a ellos en diferentes grados. Con reglas difusas, es posible procesar las relaciones entre las variables fuzzy y producir una salida nebulosa.

- Tratamiento personalizado: Basándose en las características aprendidas de los habitantes de la casa, podría darse un trato personalizado a cada uno. Tal vez el razonamiento cuantitativo sea mejor interpretado por personas con una mejor formación educativa, pero está claro que un niño de 9 años puede comprender mejor términos como mayor, caluroso, oscuro o rápido que términos numéricos. Del mismo modo, las cualidades varían dependiendo de los usuarios. El término joven será entendido de una manera por un anciano y de otra por un adolescente.

### 3 Conclusiones y trabajo futuro

Tras haber comentado la arquitectura y los servicios más importantes del proyecto Domoweb, observamos una serie de carencias sobre las que estamos trabajando para mejorar el producto final.

Una de las posibles líneas de trabajo futuro y de investigación es la de dotar de un aprendizaje automático mediante razonamiento cualitativo al sistema. Actualmente estamos desarrollando la integración de una serie de componentes que permitan esto sin tener que modificar la estructura creada. Si conseguimos que nuestro proyecto lleve a cabo el aprendizaje de forma autónoma, de acuerdo a las necesidades del usuario y además responda a los deseos y necesidades de los mismos de manera automática, habremos dado un paso importante hacia la Inteligencia Ambiental. La anticipación ante actos repetitivos mediante la base de conocimiento del sistema debe ser una realidad y existen otros trabajos que lo avalan [3]. Además, nuestro sistema debe ser lo menos intrusivo posible, evitando que los habitantes del hogar se sientan observados por los sensores del sistema.

### Agradecimientos

Este artículo ha sido financiado por el proyecto RAITEC "Domoweb: Metodologías para el diseño y desarrollo de sistemas domóticos controlados vía Web". Financiado por la Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía. Cuyo expediente es el SC/AT/0108/2003 y cuyo código de Proyecto es 2001001216. Investigador Principal: Juan Antonio Ortega Ramírez – Depto. Lenguajes y Sistemas Informáticos de la Universidad de Sevilla

### Referencias

- [1] Álvarez-Sánchez, J.J., Álvarez-Bravo, J.V., González-Cabrera, F.J., "Aproximación cualitativa a la docencia de conceptos físicos". SPDECE 04. Guadalajara (España). Octubre 2004.
- [2] BISC <http://www-bisc.cs.berkeley.edu/>
- [3] Doctor F., Hagrais, H. A. K., Callaghan, V. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans. An Intelligent Fuzzy Agent Approach to Realising Ambient Intelligence in Intelligent Inhabited Environments. 2004.
- [4] FuzzyJ. [http://ai.iit.nrc.ca/IR\\_public/fuzzy/fuzzyJToolkit.html](http://ai.iit.nrc.ca/IR_public/fuzzy/fuzzyJToolkit.html)
- [5] Hall, R. Oscar --Bundle Repository. <http://oscar.osgi.sourceforge.net>
- [6] Hall, R. Oscar, An OSGi framework implementation, <http://oscar.objectweb.org>
- [7] JDBC API. <http://java.sun.com/products/jdbc>
- [8] jUSB. Java API for USB. <http://jusb.sourceforge.net.2003>.
- [9] Motorola. Wireless Software, Applications & Services. Java APIs for Bluetooth Wireless Technology (JSR-82). 2002.
- [10] Open Source Gateway Initiative. OSGi. 1999.
- [11] OSGi Alliance. "OSGi Service Platform Release 3". 2003.
- [12] Parra, X., Angulo, C., Agell, N., Rovira, X., "Aproximación a un problema financiero mediante redes neuronales con funciones base radiales y máquinas de soporte vectorial". Inteligencia Artificial, (14): 54-67. ISSN: 1137-3601. 2001.
- [13] The Knoplerfish Project. <http://knoplerfish.org/index.html>
- [14] Weiser, M. The computer for the 21st Century. Scientific American, 1991.
- [15] X10 – Interface Communication Protocol. [http://www.maison-domotique.com/telechargements/x10/protocol\\_X10.pdf](http://www.maison-domotique.com/telechargements/x10/protocol_X10.pdf) . 2003.
- [16] Zadeh A.L. Fuzzy Sets, Inf. Control 8, 338-353, 1965.