

# El proyecto Interact: El rol de la información contextual

Pablo A. Haya, Xavier Alamán y Germán Montoro  
Dpto. de Ingeniería Informática. Universidad Autónoma de Madrid.  
Ctra. Colmenar Viejo. km 15. Madrid 28049. SPAIN  
{Pablo.Haya, Xavier.Alaman, German.Montoro}@ii.uam.es

**Palabras clave:** Computación ubicua, información contextual, pizarra.

**Resumen.** Se presenta una propuesta para conseguir sinergia entre los distintos componentes dentro de un entorno inteligente. Cada componente produce información contextual que se publica en una pizarra central accesible por todos los elementos del sistema. Esta pizarra representa un modelo de entorno, y es usada por todos los elementos activos para compartir información e interactuar entre sí. Se ha desarrollado un prototipo sobre un entorno real en el cual se incluye un modo de interacción mediante lenguaje natural.

## Introducción: Computación ubicua y entornos inteligentes

Uno de los primeros esfuerzos dentro de la computación ubicua se realizó en el Electronics and Imaging Laboratory perteneciente al Xerox Palo Alto Research Centre a finales de los ochenta. En 1988, en este laboratorio se desarrolló el programa de Ubiquitous Computing, que incluía los proyectos LiveBoard [ELRO92], ParcPad [KANT93] y ParcTab [WANT95]. Durante los siguientes años se desarrolló el conocido sistema Active Badge System [WANT92] (original de Olivetti) y en 1992 se implementó el primer sistema "ubi-comp" [WEIS99].

El término de "computación ubica" fue definido por primera vez por Mark Weiser [WEIS91]. De acuerdo con Weiser, las dos características más importantes son la ubicuidad y la transparencia del sistema. Para satisfacer estos requerimientos los dispositivos tienen que ser móviles o deben estar integrados dentro del entorno (una habitación, una oficina, un edificio etc.).

Actualmente varios centros de investigación están realizando importantes aportaciones dentro de la computación ubicua [SATA01]. El proyecto Oxygen se está desarrollando en el laboratorio de Computer Science and Artificial Intelligence del MIT [DERT99]. El proyecto Aura, del Instituto Human Computer Interaction perteneciente a la CMU, es una plataforma para facilitar la movilidad del usuario dentro de entornos ubicuos [FLIN01]. Otro proyecto relevante es "Portolano", del Departamento de Computer Science & Engineering de la Universidad de Washington [ESLE99]. Dentro de la industria se destacan los trabajos de AT&T Research en Cambridge, Reino Unido. [<http://www.uk.research.att.com/projects.html>] y del IBM TJ Watson Research Center [<http://www.research.ibm.com>].

Un enfoque que cabe destacar es el defendido por Alex Petland [PENT00], que propugna por una computación ubica en la que cada dispositivo sea una aplicación aislada y capaz de funcionar por sí misma, minimizando la comunicación entre los componentes del entorno.

Uno de los paradigmas en los cuales se puede aplicar la computación ubicua es el de los entornos inteligentes. Un entorno inteligente consiste en un conjunto de entidades con capacidad de computación compartiendo un mismo espacio físico, y que son capaces de interactuar con el mundo físico, sus habitantes y entre ellas. Dicho con otras palabras es un "espacio interactivo y altamente integrado, que acerca la computación al mundo real". Esto permite a los ordenadores "participar en actividades que nunca antes habían podido" y a las

personas “interaccionar con el sistema computacional mediante los mismo medios que utilizan para comunicarse entre si: gestos, voz, movimiento y contexto” [COEN98].

Algunas de las características de un entorno inteligente fueron definidas por primera vez por Alex Pentland [PENT96]. Entornos que son capaces de ver y oír, lo cual les permite localizar a los usuarios, saber quiénes son, qué están haciendo, y por qué. Pueden interaccionar con el usuario de una manera intuitiva (por voz, visión, gestos, escucha...) y aprender patrones de interacción que mejoren la calidad de comunicación con el usuario [NIXO99]. Un decálogo de los requerimientos que debería tener un entorno inteligente se puede encontrar en Shafer [SHAF99].

Alguno de los proyectos que actualmente se están desarrollando en entornos inteligentes incluyen a instituciones como el MIT, el INRIA, Georgia Tech o el Microsoft Research Center. En el Artificial Intelligence Laboratory del MIT se está desarrollando el proyecto Intelligent Room cuyo objetivo es integrar elementos computacionales en el entorno de forma que los usuarios puedan interaccionar con ellos mediante la voz, gestos, y contexto, incorporando los ordenadores a la vida diaria [GAJO01]. El proyecto Prima del laboratorio Graphics Vision y Robotics del INRIA Rhône-Alpes utiliza tecnologías para el diseño e integración de sistemas que permitan percibir y reconocer a las personas y sus acciones dentro de un entorno inteligente [LEGA01]. La Aware Home Research Initiative del grupo Future Computing Environments perteneciente al Georgia Tech persigue construir un entorno inteligente dentro de una casa real [KIDD99]. Por último el proyecto EasyLiving del grupo Microsoft Research Vision Group está desarrollando tecnología básica para la construcción de entornos inteligentes [BRUM00].

En España dentro del campo de la computación ubicua se han realizado importantes progresos, tanto en entornos educativos [ORTE01], como en museos virtuales y espacios naturales [SEND01], plataformas de agentes móviles [CAMP02], y tecnologías de la rehabilitación [ABAS02].

### **INTERACT: una propuesta para gestión del contexto**

En este artículo se propone la capa de *contexto* como pegamento para conseguir la sinergia necesaria entre los elementos ubicuos que constituyen el entorno inteligente. De acuerdo con otros trabajos tales como [GROSS01], creemos que un modelo del mundo centralizado, aunque implementando de forma distribuida, es la mejor opción para obtener interacciones complejas entre todos los dispositivos.

La capa de contexto aporta:

1. Una única interfaz que abstrae los detalles de comunicación entre los diferentes dispositivos físicos.
2. Un modelo unificado del mundo.

La implementación de la capa de contexto reside en una estructura de datos global denominada pizarra [ENGE88]. Esta pizarra es un modelo del mundo, donde se encuentra almacenada toda la información importante relacionada con el entorno (incluyendo sus usuarios). La pizarra almacena también información sobre los flujos de información entre los dispositivos multimedia (micrófonos, altavoces, cámaras, pantallas ...).

Los dispositivos ubicuos utilizan la información de la pizarra para entender el contexto y adaptarse a él. Por ejemplo, el número de personas en la habitación, la tarea que están realizando en cada momento, y el estado de los dispositivos físicos se almacenan en la pizarra.

## ¿ Qué entendemos por contexto ?

El contexto se puede definir como el estado del entorno en un instante dado. En los entornos inteligentes, una misma acción del usuario puede producir consecuencias diferentes dependiendo del contexto. Cada entidad que conforma el entorno tiene asociada una porción de contexto. De esta forma, el estado del entorno emerge a raíz del estado particular de todas las entidades involucradas.

Cada fragmento de contexto se representa como un conjunto de propiedades, donde una propiedad es par nombre-valor. El valor de una propiedad se puede medir directamente del entorno, o se puede deducir a partir de los valores de otras propiedades. Por ejemplo, una propiedad puede ser el número de personas de una habitación, que puede ser calculada analizando las imágenes obtenidas por una video cámara. El contexto del entorno es la suma de las propiedades medidas directamente y de las deducidas.

### Fuentes de información contextual

Una fuente de información contextual es cualquier entidad computacional que pertenece al entorno. Dentro de un entorno inteligente se pueden encontrar fuentes de información contextual de muy distinta naturaleza. Las fuentes pueden ser muy cercanas al mundo físico, o por el contrario puede estar relacionadas con el mundo virtual. El primer conjunto se compone de toda clase de dispositivos que interaccionan con el mundo físico tales como, sensores, conmutadores, electrodomésticos, pantallas, micrófonos, altavoces, etc. El segundo conjunto incluye aquellos componentes puramente computacionales, tales como gestores de diálogos, agente inteligentes, clientes de correo, etc. Esta mezcla heterogénea de fuentes hardware y software implica ciertos desafíos:

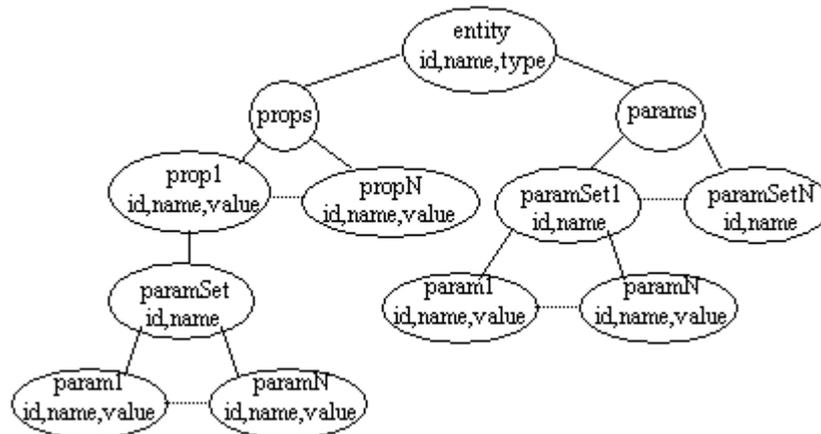
- Se han de integrar y gestionar diferentes tecnologías.
- Las fuentes de información esta muy distribuidas.
- Cada componente aporta un nivel de abstracción distinto. Por ejemplo, un sensor maneja información de alta resolución pero poco abstracta.

Nuestra propuesta trata de solucionar estos requerimientos. Para ello nos basamos en una representación estándar del conocimiento e independiente de su naturaleza y nivel de abstracción, y en un mecanismo común para poder acceder a esa información.

### Pizarra contextual

Nuestro principal requisito es encontrar una estructura de datos que pueda representar tanto el estado de una entidad individual como de todo el entorno. Se ha elegido una estructura de tipo pizarra, la cual almacena la información contextual de cada entidad que pertenece al entorno. Para representar una entidad, se ha elegido un mecanismo general y flexible. Cada entidad se define por un conjunto de pares nombre-valor. Cada par puede aporta dos tipos distintos de información. Por una parte, se pueden encontrar propiedades que modelan características intrínsecas de la entidad, es decir, aquellas que son universalmente aceptadas. Por otro lado, se pueden encontrar los parámetros, que se definen localmente para cada aplicación, o para un grupo de aplicaciones. Los parámetros se organizan en conjuntos, de tal forma que cada aplicación puede definir sus propios parámetros independientemente del resto. Cada entidad se identifica con un número que es único para todo el sistema, y una cadena de caracteres, la cual puede repetirse. Además, cada entidad se le asigna un tipo único (una entidad puede ser

un dispositivo, una persona, una habitación ...). En la Figura 1 se muestra gráficamente la representación en árbol de cada entidad.



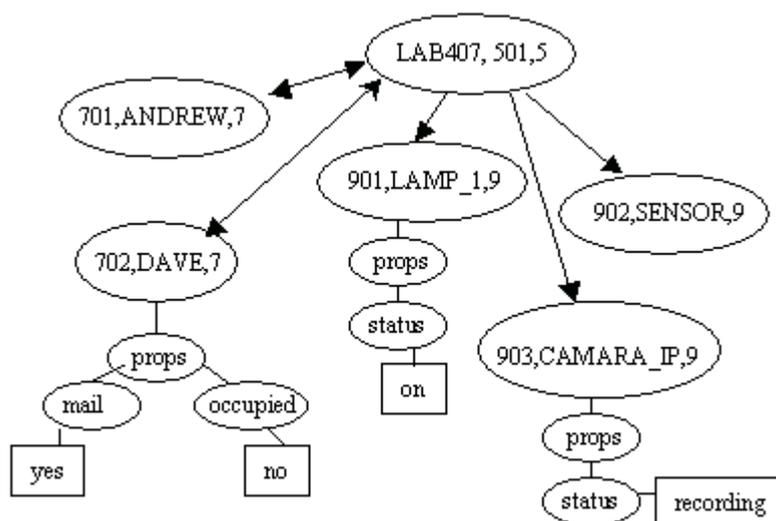
**Figura 1. Representación en forma de árbol de una entidad**

La pizarra almacena además de la información de contexto de cada entidad, la relaciones entre las entidades. Una relación puede ser de cualquier tipo (asociación, agregación ...) y de cualquier dirección (unidireccional o bidireccional). Por tanto, la pizarra se compone por un grafo de entidades, donde cada entidad es un árbol.

### Árboles de contexto

Aunque la estructura general de la pizarra es un grafo, la información que se intercambia entre la pizarra y los agentes se estructura utilizando árboles de contexto. Estos son representaciones parciales del grafo que constituye toda la pizarra. Cada árbol se compone de una raíz, y un conjunto de nodos, lo cual permite organizar jerárquicamente tanto la relación entre las entidades como la información, en forma de propiedades y parámetros, perteneciente a cada entidad. De esta forma cada árbol de contexto aporta información sobre el estado interno de una entidad, y sobre sus relaciones dentro de la pizarra.

Cada nodo del árbol puede ser una entidad, una propiedad o un parámetro. Independientemente de su naturaleza tendrá: un identificador, un nombre, un valor, un conjunto variable de nodos hijo, y un único padre.



**Figura 2. Pizarra Contextual: entidades y sus relaciones**

Esta estructura gestiona la información siguiendo distintos niveles de abstracción. Los nodos más profundos representan información más concreta, mientras que los nodos más cercanos a la raíz reflejan las relaciones entre los componentes (ver Figura 2). Un árbol de contexto se puede entender como una instantánea de una parte del estado del entorno que se genera dinámicamente.

Cada nodo que pertenece a la pizarra tiene uno o más identificadores que permiten nombrarlo unívocamente respecto al resto. Se puede acceder a un nodo mediante los siguientes mecanismos:

1. Usando una jerarquía predefinida. Cada jerarquía define el orden en que se recorren los nodos, y se identifica con una cadena de caracteres. Por ejemplo, siguiendo con la Figura 2 el *nid* /roomdevice/lab407/lamp\_1/props/status correspondería a una jerarquía denominada como *roomdevice* que establece que el primer nodo es de tipo habitación, y el segundo nodo de tipo dispositivo. Los restantes nodos corresponderían a propiedades o parámetros del dispositivo. Cada jerarquía se define estáticamente y cada aplicación puede definir su propia jerarquía. Las jerarquías disponibles en tiempo de ejecución han sido predefinidas en tiempo de desarrollo. Esto permite conseguir una alta eficiencia en el sistema sin perder la flexibilidad que sea necesaria por parte de los distintos agentes.
2. Existe siempre una jerarquía por defecto que emplea el tipo de la entidad como comienzo del *nid*. Por ejemplo, si queremos referenciar la propiedad que almacena si *dave* tiene o no correo electrónico, se puede aprovechar la información de que *dave* es una persona, así quedaría el *nid* /people/dave/props/mail.
3. Todos los nodos se pueden localizar a partir del identificador numérico de una entidad. Para ello el *nid* empezará con la palabra clave *id* seguido del identificador de la entidad. A continuación vendrán los identificadores necesarios para llegar al nodo deseado. Por ejemplo, siguiendo el árbol de la Figura 2, el *nid* del nodo *status* sería /id/901/props/status.

Además, en cualquier parte del *nid* se pueden utilizar comodines para referenciar más de una entidad, propiedad o parámetro. Esto permite acceder a varios nodos a la vez. Por ejemplo el *nid* /roomdevices/lab407/\* referencia a todos los dispositivos dentro del laboratorio b407.

## Interacción con la pizarra

La comunicación entre los dispositivos, la pizarra y los agentes se basa en una arquitectura de tres capas, compuesta por la capa física, la capa de contexto, y la capa de aplicación. La primera incluye a aquellos componentes que obtienen medidas directamente del mundo real, mientras que la capa de aplicación la forma agentes inteligentes que generan propiedades deducidas del mundo real, o propiedades relacionadas con el mundo virtual. La capa intermedia sirve como pegamento entre las otras dos capas.

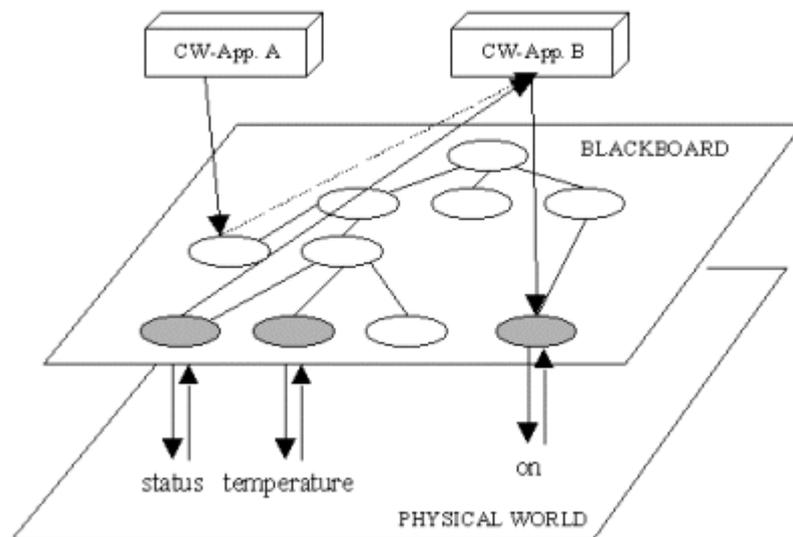
En nuestra propuesta, la información contextual generada se publica en un almacén central accesible para todo el sistema, siguiendo el paradigma clásico de pizarra.

Cada pizarra contiene información sobre todas las entidades del entorno, sus propiedades y parámetros, y sus relaciones. Mediante un protocolo estándar se puede solicitar o modificar los valores de los nodos, además de poder subscribirse a los cambios producidos. Los agentes pueden acceder fácil y transparentemente al valor de las propiedades y los parámetros, de forma que la naturaleza de la fuente de información contextual queda oculta. Por ejemplo, una propiedad del contexto de una habitación puede ser la tarea que está realizando un usuario. Se utilizarán varias fuentes de información para deducir la tarea, pero en la pizarra se almacenará un único valor accesible independiente de la naturaleza de la fuente.

El proceso de interacción se puede resumir como sigue: las fuentes de información contextual envían los cambios en el contexto a la pizarra, y en consecuencia la pizarra

modifica el valor de los nodos. Los agentes pueden enterarse de esos cambios, o bien preguntando a la pizarra, o bien mediante una suscripción previa. De esta forma, la pizarra actúa como intermediario, almacenando la información contextual. Los componentes de las otras dos capas son los encargados de procesar los cambios, y actuar en consecuencia.

Por ejemplo, si alguien entra en una habitación vacía, el agente de visión alerta del nuevo estado modificando el valor de una propiedad de la pizarra. Otro agente suscrito a los cambios de ese nodo se entera de la modificación. A continuación comprueba el valor de otra propiedad de la pizarra que indica la luminosidad de la habitación. Si se encuentra demasiado oscura, el agente ilumina la habitación modificando el valor de una propiedad de la pizarra que aumenta la luz de la habitación. El componente de la capa física encargado de regular la luz reacciona al cambio en la pizarra ajustando la luminosidad al nuevo estado.

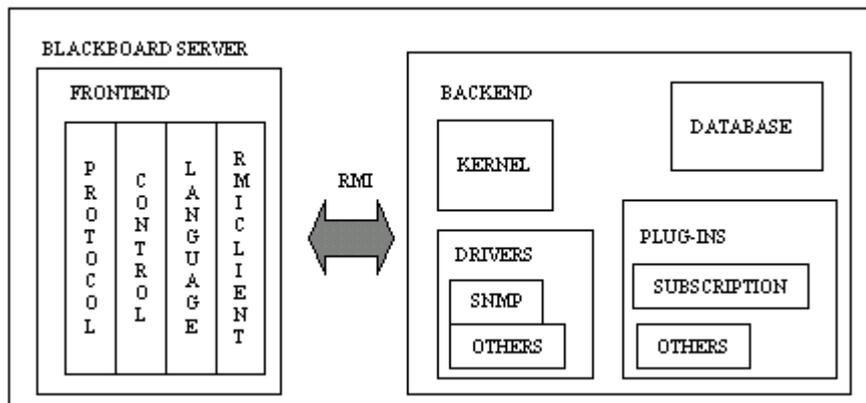


**Figura 3. Proceso de interacción con la pizarra**

Las propiedades que se corresponden directamente con el mundo real se tratan de una forma especial. En estos casos, el valor de la propiedad no se encuentra almacenado en la pizarra, sino que ésta actúa como si fuera un *proxy*. Siempre que se pide el valor de la propiedad, la pizarra redirecciona la pregunta al dispositivo físico. Por otro lado, si el dispositivo modifica su estado, la pizarra se encarga de que el suceso se notifique a todos los suscriptores. La Figura 3 muestra cómo se realiza el proceso completo de interacción. Las líneas discontinuas representan notificaciones de cambios en el contexto, y las líneas sólidas peticiones de agentes para preguntar sobre el estado de un nodo, o para cambiarlo. Los nodos oscuros representan propiedades que no se almacenan en la pizarra directamente.

## Implementación

Nuestra primera implementación se realizó utilizando la Context Toolkit [DEY99]. Actualmente, se ha desarrollado un servidor de pizarra propietario. La arquitectura del sistema se muestra en la Figura 5.



**Figura 4. Arquitectura del prototipo de pizarra**

La pizarra tiene dos componentes principales:

- *Backend*: consta de los siguientes módulos. (a) Núcleo, este módulo implementa la funcionalidad básica de la pizarra, en concreto las operaciones *GetContext*, *SetContext*, *AddContext* y *DeleteContext* (b) Una base de datos, que almacena las entidades, las propiedades, los parámetros, y las relaciones entre entidades (c) *Plug-ins*: que implementan funcionalidad adicional que puede ser integrada sin modificar el resto de la arquitectura. Todas las pizarras incluyen al menos el módulo de suscripciones. (d) Un conjunto de controladores, que permite comunicar a la pizarra con los diferentes dispositivos del entorno.
- *Front-end*: esta parte implementa la conexión y comunicación con el cliente. Se pueden añadir varios *front-ends* a la arquitectura, cada uno con un protocolo y un lenguaje de representación distintos. En la actualidad se soporta HTTP, como capa de transporte y XML, como lenguaje de intercambio de información.

Se ha desarrollado un prototipo de entorno inteligente utilizando la capa de contexto descrita en apartados anteriores. Este prototipo consiste en una habitación equipada como si fuera el salón de una casa moderna. (ver Figura 5).

En el prototipo se utilizan dos redes de comunicaciones distintas. Para la conexión de sensores (presencia, temperatura, luminosidad, etc.) y los actuadores (interruptores, reguladores, etc.) se utiliza el bus europeo EIB [<http://www.eiba.com>]. Este bus trata de convertirse en un estándar dentro de la domótica en la Unión Europea. Para los datos multimedia (micrófonos, cámaras, etc.), se emplea una red Ethernet (tanto cableada como inalámbrica) que permite conectar los diferentes dispositivos. Cada dispositivos se conectará a una de las dos redes dependiendo de su naturaleza, y puede ser controlado a través de la pizarra. Para facilitar la comunicación entre la pizarra y los dispositivos se emplea el protocolo SMNP (Simple Management Network Protocol), de forma que independientemente si es un dispositivo EIB, o multimedia se controla con el mismo protocolo [MART03].

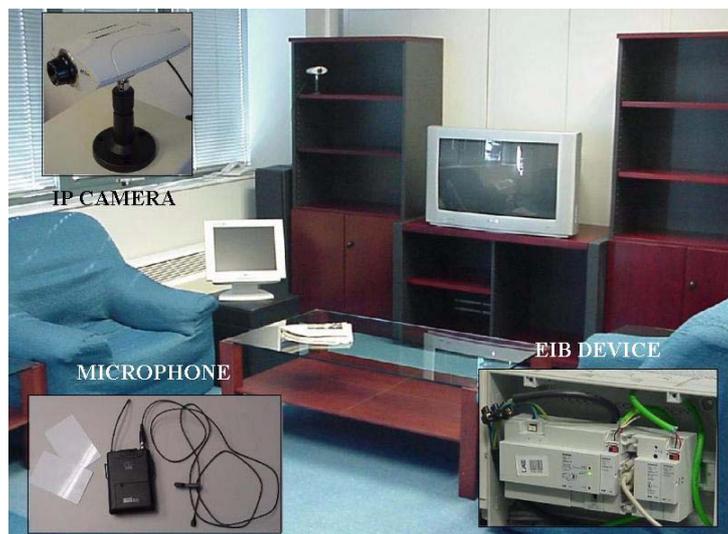
Los dispositivos instalados se dividen en tres categorías:

- *Domótica*: se compone de varios sistemas independientes: portero electrónico que controla la apertura de la puerta, el sistema de tarjetas inteligentes que informa de quién está entrando en la habitación, y varios dispositivos EIB, tales como relés electrónicos para controlar las luces, interruptores, pantallas alfanuméricas, etc.
- *Información audio-visual*: incluye una televisión, dos altavoces hi-fi, varios altavoces normales, una radio, un DVD, y varias pantallas planas. Actualmente se puede disponer de la salida de audio en cualquier altavoz de la habitación simplemente cambiando en la

pizarra la distribución de los flujos de audio. También se puede conmutar la música de la habitación entre un canal de radio, y un CD-ROM, y controlar el volumen. Por último, una cámara de video IP permite monitorizar la actividad de la habitación.

- *Interacción vocal:* varios micrófonos inalámbricos permiten interactuar con un gestor de diálogos, permitiendo al usuario que se mueve libremente por la habitación.

El prototipo se completa con la pizarra anteriormente expuesta en la cual se incluye la definición de todos los dispositivos descritos.



**Figura 5. Instantánea del entorno**

Finalmente, se dispone de un agente de diálogos que se ha integrado en el entorno, que permite realizar un control básico de los dispositivos de la casa empleando lenguaje natural.

### **El agente de diálogo: un primer ejemplo de agente de contexto**

Uno de los puntos más importante del entorno inteligente que se está desarrollando es la interfaz vía voz. Se ha implementado un sistema de diálogos robusto que permite al usuario controlar los dispositivos de la habitación utilizando lenguaje natural. La información contextual necesaria para guiar el diálogo se adquiere de la pizarra. La implementación se basa en varios módulos de dialogo independiente, de forma que cada módulo se encarga de un tema concreto. Un gestor de diálogos, en base a la información del entorno y al discurso del usuario, decide cual es el diálogo activo en cada momento.

Se puede acceder a la mayor parte de las entidades almacenadas en la pizarra mediante diálogos específicos. El gestor de diálogos interactúa con la pizarra igual que cualquier otro agente, utilizando los comandos disponibles. De esta forma es capaz de obtener información y modificar el entorno en tiempo de ejecución.

El tener la información contextual almacenada y fácilmente accesible en la pizarra es la clave para desarrollar un gestor de diálogos robusto. El desarrollador conoce a priori toda la información de la que dispone, además puede utilizar el contexto para resolver frases incomprensibles. Por ejemplo, si el usuario dice “*por favor, ¿ puedes apagar la radio ?*”, y el reconocedor no entiende la palabra “radio”, el gestor puede preguntar por los dispositivos encendidos, y si solo está la radio, apagarla directamente.

La implementación de los diálogos se basa en plantillas, similares a las descritas por Schank en [SCHA77]. Actualmente, se está investigando la generación automática de

diálogos, de tal forma que se puedan crear o modificar las plantillas en tiempo de ejecución utilizando información almacenada en la pizarra.

## **Conclusión y trabajo futuro**

En este artículo se ha presentado un proyecto multidisciplinar en el cual se intenta integrar distintos componentes computacionales dentro de un entorno inteligente. Se propone una capa denominada de contexto, donde se almacena la información del entorno, para conseguir la interacción entre el conjunto heterogéneo de dispositivos y agentes que conforman un entorno inteligente. En este momento, se dispone de un prototipo funcionando que incluye las tres capas descritas en la arquitectura: la capa física (integrada por una red domótica EIB y una red TCP/IP para el tráfico multimedia, así como varios dispositivos físicos), la capa de contexto, y un agente de diálogos que utiliza la información contextual.

Los usuarios interaccionan con la habitación utilizando micrófonos inalámbricos. Debido a que esta no es la forma más cómoda para el usuario, se está trabajando en la eliminación de los micrófonos. Para ello se están estudiando distintas técnicas: como el uso distintos reconocedores de forma redundante (HTK, Dragon Naturally Speaking, etc.) y el uso de arrays de micrófonos. El estudio de la eficacia de las diversas técnicas será uno de los siguientes objetivos del proyecto.

Se está desarrollando un agente de visión que permita realizar un seguimiento de usuario, y deducir la tarea que está realizando. Actualmente se dispone de un agente de visión que permite detectar movimiento en la habitación.

Por último, se están planteando varias reformas en la arquitectura. En primer lugar, almacenar el histórico de los cambios en los valores de una propiedad, de forma que se puedan realizar análisis posteriores. En segundo lugar, incorporar un mecanismo de caché a la pizarra para evitar accesos innecesarios a los dispositivos físicos. Por último, se está diseñando un mecanismo de seguridad que restrinja el acceso a la información de la pizarra a agentes autorizados.

Este proyecto está financiado por el Ministerio de Ciencia y Tecnología, con número de proyecto TIC2000-0464.

## **Referencias y Bibliografía**

[ABAS02] J. Abascal, D. Cagigas, N. Garay, L. Gardeazabal: Mobile Interface for a Smart Wheelchair. Fourth International Symposium on Human Computer Interaction with Mobile Devices. Pisa (Italy), 2002. In F. Paternò (ed.) Human Computer Interaction with Mobile devices. LNCS 411, Springer, Berlin 2002. ISBN 3-540-44189-1. Pp 373-377

[BRUM00] Brumitt, B.L.; Meyers, B.; Krumm, J.; Kern, A. and Shafer, S. EasyLiving: Technologies for Intelligent Environments. Handheld and Ubiquitous Computing, 2nd Intl. Symposium, September (2000), pp. 12-27.

[CAMP02] Campo, C. Agentes móviles en computación ubicua. Actas del Tercer Congreso Interacción Persona-Ordenador. Interacción, Madrid (2002).

[COEN98] Coen, M.H. Design Principles for Intelligent Environments. In Proceedings of the AAAI Spring Symposium on Intelligent Environments (1998).

[DERT99] Dertouzos, M.L. The future of computing. Scientific American, July (1999).

[DEY99] Dey, A. K. et al. A Context-Based Infrastructure for Smart Enviroments. In Proceedings of the Managing Interactions in Smart Enviroments (1999).

[ELRO92] Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., Tang, J. and Welch, B. Liveboard: a large interactive display supporting group meetings, presentations and remote collaboration. In Proceedings of the CHI'92 Conference on Human Factors in Computing Systems, New York (1992) pp .599-607.

- [ENGE88] Englemore, R. And Morgan, T. *Blackboard Systems*, Addison-Wesley, (1988).
- [ESLE99] Esler, M.; Hightower, J.; Anderson, T. and Borriello, G. Next Century Challenges: Data-Centric Networking for Invisible Computing: The Portolano Project at the University of Washington. Proceedings of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, August 15-19, Seattle, Washington, USA, (1999) pp. 256-262
- [FLIN01] Flinn, J., Narayanan, D., and Satyanarayanan, M. 2001. Self-tuned remote execution for pervasive computing. In Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), Schloss Elmau, Germany, May (2001)
- [GAJO01] Gajos, K.; Weisman, L. and Shrobe, H. Design Principles For Resource Management Systems For Intelligent Spaces. In Proceedings of The Second International Workshop on Self-Adaptive Software. Budapest, Hungary, (2001).
- [GROS01] Grossman, M., Leonhardi, A., Mitschang, B., and Rothermel, K. A World Model for Location-Aware Systems, *Informatique-Revue des Organisations Suisses d'Informatique*, Special issue on Ubiquitous Computing, No. 5, october (2001).
- [KANT93] Kantarjiev, C.K., Demers, A., Krivacic, R.T., Frederick R. and Weiser, M. 1993. Experiences with X in a wireless environment. In Proceedings USENIX Symposium on Mobile & Location-independent Computing. USENIX Association, August (1993) pp. 117-128
- [KIDD99] Kidd, C.D.; Orr, R.; Abowd, G.D.; Atkenson, C.G.; Essa, I.A.; Macintyre, B.; Mynatt, E.; Starner, T.E. and Newstetter, W. The Aware Home: A Living Laboratory for Ubiquitous Computing Research. In Proceedings of the Second International Workshop on Cooperative Buildings (1999).
- [LEGA01] Le Gal, C.; Martin, J.; Lux, A. and Crowley, J. L. SmartOffice: Design of an Intelligent Environment. *IEEE Intelligent Systems*, 16, 4 (2001).
- [MART03] Martínez, A.E., Cabello, R., Gómez, F.J. and Martínez, J. *Interact-DDM: A Solution for the Integration of Domestic Devices on Network Management Platforms*. IFIP/IEEE International Symposium on Integrated Network Management Colorado Springs, Colorado, USA. (2003)
- [NIXO99] Nixon, P.; Dobson, S. and Lacey, G. Smart Environments: some challenges for the computing community. In Proceedings of 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99, Dublin, Dec 1999), P. Nixon, G. Lacey and S. Dobson, Eds. Springer-Verlag, London, (1999) pp. 1-4.
- [ORTE01] Ortega, M.; Redondo, M. A.; Paredes, M.; Sánchez-Villalón, P. P.; Bravo, C. y Bravo, J. Nuevos paradigmas de interacción en el aula del siglo XXI. II Congreso Internacional de Interacción Persona-Ordenador, Salamanca (2001) pp. 161-171.
- [PENT00] Pentland, A. Perceptual Intelligence. In *Communications of the ACM*. Vol. 43, No 3, March (2000).
- [PENT96] Pentland, A. Smart rooms. *Scientific American*, 274, 4, (1996) pp. 68-76.
- [SATA01] Satyanarayanan, M. Pervasive Computing: Vision and Challenges. *IEEE Personal Computing*. August (2001) pp. 10-17.
- [SCHA77] Schank, R. and Abelson, R. *Scripts, Plans and Goals*. Erlbaum, Hillsdale, NJ. (1977)
- [SEND01] Sendín, M. ; Lorés, J.; Aguiló, C. y Balaguer, A. Un modelo interactivo ubicuo aplicado al patrimonio natural y cultural del área del Montsec. *Revista Novática*. 153 (2001) pp. 22-25.
- [SHAF99] Shafer, S. Ten dimensions of ubiquitous computing. In Proceedings of 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99, Dublin, Dec 1999), P. Nixon, G. Lacey and S. Dobson, Eds. Springer-Verlag, London, (1999) pp. 5-16.
- [WANT92] Want, R.; Hopper, A.; Falcao, V. And Gibbons, H. The active badge location system. *ACM Transactions on Information Systems*, 10, 1, (1992) pp. 91-102.
- [WANT95] Want, R., Schilit, B.N., Adams, N.I., Gold, R., Petersen, K., Goldberg, D., Ellis, J.R., Weiser, M. An overview of the ParcTab ubiquitous computing experiment. *IEEE Personal Communications Magazine*. (1995) pp. 28-43.
- [WEIS91] Weiser, M. The computer of the 21st century. *Scientific American*, 265, 3, (1991) pp. 66-75.
- [WEIS99] Weiser, M.; Gold, R. and Brown, J. S. The origins of ubiquitous computing research at PARC in the late 1980s. *IBM Systems Journal*, 38, 4, (1999) pp. 693-696.  
24-28 de Marzo, 2003.