

UNIVERSIDAD AUTÓNOMA DE MADRID

Arquitecturas para el desarrollo de Entornos Inteligentes

Autor:

Pablo A. Haya Coll

Pablo.Haya@ii.uam.es

Tutor:

Xavier Alamán Roldán

Xavier.Alaman@ii.uam.es

junio de 2001

DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

Índice

Índice	I
Índice de Tablas.....	III
Índice de Figuras	IV
Introducción.....	1
Capa física	5
2.1 Redes de datos	7
2.1.1 Redes de control	7
2.1.1.1 EIB (European Installation Bus).....	8
2.1.1.2 X10	15
2.1.1.3 LONWORK.....	16
2.1.1.4 CEBus.....	18
2.1.1.5 NUDAN.....	19
2.1.1.6 Konnex.....	20
2.1.1.7 BACNet	21
2.1.2 Redes Multimedia.....	21
2.1.2.1 IEEE 1394 (FireWire)	22
2.1.2.2 USB (Universal Serial Bus).....	23
2.1.2.3 HomePNA	24
2.1.2.4 Media Wire	24
2.1.3 Red Ethernet	25
2.1.4 Redes Inalámbricas.....	27
2.1.4.1 IEEE 802.11.....	30
2.1.4.2 HomeRF.....	31
2.1.4.3 BlueTooth	32
2.2 Conclusiones.....	33
Capa Intermedia.....	37
3.1 Capas intermedias comerciales.....	38
3.1.1 Jini	38
3.1.2 UPnP (Universal Plug & Play).....	40
3.1.3 HomeAPI.....	42
3.1.4 HAVi (Home Audio Visual interoperability).....	43
3.1.5 OSGI (Open Service Gateway Initiative).....	44
3.1.6 SCP (Simple Control Protocol) – SoftWire.....	45
3.2 Capas intermedias en proyectos de investigación	46
3.2.1 Context Toolkit.....	46
3.2.2 MetaGlue	49
3.2.3 Smart Office	51
3.2.4 AutoHan.....	52
3.2.5 HomeNet.....	53
3.3 Conclusiones.....	54
Arquitectura del proyecto InterAct.....	59
4.1 La capa de interacción con el entorno físico	60
4.1.1 Red de Control.....	61
4.1.2 Red Multimedia	61
4.2 La capa de contexto	62
4.2.1 Arquitectura	62
4.2.2 Flujos de información.....	63

4.3 Primer prototipo.....	64
4.4 Conclusiones.....	65
Conclusiones y trabajo futuro.....	67
5.1 Conclusiones.....	67
5.2 Trabajo futuro.....	70
Agradecimientos.....	73
Bibliografía.....	75
Apéndice I.....	79
Apéndice II.....	83

Índice de Tablas

Tabla 1. Clasificación en función del cableado de la red	7
Tabla 2. Clasificación de diferentes protocolos de control en función de topología, control y modelo de comunicación.....	8
Tabla 3. Número de dispositivos LONWork permitido de forma desglosada	17
Tabla 4. Clasificación de distintos tipos de tráfico multimedia según el ancho de banda requerido	22
Tabla 5. Relación de los distintos estándares Ethernet con los posibles medios físicos	25
Tabla 6. Comparación de distintas tecnologías aplicables a una red LAN según la cuota de mercado y la velocidad de transmisión. Fuente: [Orfali,R. et al. 1999]	27
Tabla 7. Clasificación de distintas red inalámbricas	30
Tabla 8. Comparación de distintos protocolos de comunicación con un PC. Fuente: [Held,G. 2000]	34
Tabla 9. Protocolos empleados por UPnP	40
Tabla 10. Clasificación de capas intermedias según su relación con el lenguaje de programación Java	56

Índice de Figuras

Figura 1. Topología de una red EIB	10
Figura 2. Formato de una dirección física EIB.....	10
Figura 3. Formato de una dirección de grupo EIB de nivel 2	11
Figura 4. Formato de una dirección de grupo EIB de nivel 3	11
Figura 5. Pulso de transmisión EIB	11
Figura 6. Formato de un telegrama EIB	12
Figura 7. Arquitectura de un dispositivo EIB.....	13
Figura 8. Clasificación de distintos tipos de tráficos multimedia según el ancho de banda requerido	24
Figura 9. Previsión para el año 2003 de las tecnologías usadas como líneas de alta velocidad en un hogar. Fuente: [Dutta-Roy,A 1999]	35
Figura 10. Componentes de un cliente UPnP	41
Figura 11. Componentes de un objeto inteligente UPnP.....	41
Figura 12. Configuración orientada ad-hoc de UPnP.....	41
Figura 13. Configuración orientada a servidor de UPnP	42
Figura 14. Arquitectura de la Context Toolkit	47
Figura 15. Arquitectura de HomeNet	53
Figura 16. Plano del cableado de red del laboratorio de entornos inteligentes	60
Figura 17. Ejemplo de pizarra utilizando la Context Toolkit.....	79

Introducción

Este trabajo de investigación se enmarca en una nueva área dentro el ámbito de las interfaces de usuario (IPO-Interacción Persona Ordenador)¹ que está comenzando a recibir una gran acogida: los así llamados “Entornos Inteligentes” (Intelligent Environments, Smart Environments).

Aunque las metáforas tradicionales que utiliza un usuario para comunicarse con un ordenador, tales como ratón, teclado y ventanas, siguen teniendo vigencia, a partir de la década de los 90 se está empezando a investigar en métodos de interacción más intuitivos y naturales. Se pretende que el usuario obtenga los servicios del sistema mediante múltiples modos de interacción (lenguaje natural, gestos, tacto, sonidos...). El usuario no tiene que acostumbrarse a una forma de interaccionar cómoda para el ordenador, sino que interactúa con él igual que si lo hiciera con otra persona, sin que le requiera un esfuerzo extra de concentración. El resultado final es que los ordenadores quedan ocultos para el usuario, integrados en el entorno.

Si el sistema, además, es capaz de percibir el estado del mundo real y es sensible al contexto, se podría hablar de Entornos Inteligentes. Éstos interactúan de forma natural con el individuo y le ayudan de manera no intrusiva en la realización de las tareas cotidianas. De este modo se consigue que las habitaciones u oficinas tengan una entidad propia y tomen la iniciativa en la interacción para mejorar la calidad de vida de las personas que las ocupan.

¹ En inglés, HCI - Human Computer Interaction

La computación ubicua como referente de los entornos inteligentes

El proyecto considerado el precursor en la investigación de entornos inteligentes surgió de la empresa Xerox Parc. a finales de los años 80. Mark Weiser [Weiser, M. 1991] expuso las bases de lo que actualmente se denomina Computación Ubicua. Weiser expone en su artículo que las barreras entre la persona y el ordenador se diluirán con la "desaparición" física de este último. Defiende que para ello es necesario dotar a la mayor parte de los objetos físicos cotidianos de capacidad de computación y de comunicación, creando una gran red de dispositivos interconectados entre sí.

Weiser y su grupo se dedicaron a crear objetos de distintos tamaños y usos siguiendo esta filosofía. Terminaron clasificando los dispositivos creados en tres grupos, según el tamaño: marcas, tabletas y pizarras. Las marcas eran pequeños objetos que siempre acompañaban a la persona. Servían para localizar al usuario en el entorno. Las tabletas tenían el tamaño de un libro o una revista. Mostraban información personalizada al usuario, aunque no tenían por qué estar vinculadas a los usuarios. Por último, las pizarras, similares a las pizarras tradicionales, difundían información audio-visual. Tanto las tabletas como las pizarras no se limitaban a presentar información, sino que contaban con lápices y tizas electrónicas que les permitían capturar la entrada del usuario. Todos los dispositivos, independientemente del tamaño, quedaban unidos mediante enlaces inalámbricos, así que la ubicuidad era total al no estar limitada por el alcance de los cables. Como resultado final, el ordenador terminaba formando parte del entorno, quedando así oculto al usuario que interactuaba con éste de manera implícita.

Actualmente las investigaciones en entornos inteligentes no se centran exclusivamente en el paradigma de computación ubicua. Han surgido nuevas aproximaciones que inciden en la idea de la desaparición del ordenador, pero planteando propuestas distintas a las de Weiser. Las dos vertientes que han tenido mayor repercusión han sido lideradas por Michael Coen [Coen, M. H. 1998] y Alex Pentland [Pentland, A. 2000].

Coen admite que los entornos inteligentes requieren una infraestructura computacional altamente integrada con el mundo real. Pero esto no implica que los elementos computacionales tengan que estar en todas partes, ni que las personas deban interactuar directamente con cualquier tipo de dispositivo. La aproximación de Coen propone realizar los menores cambios posibles en el entorno. Se basa en sustituir todos los sensores y objetos dotados de capacidades de computación por un número reducido de cámaras de vídeo y micrófonos.

Si Coen rompe con la idea de localización ubicua, Pentland acomete contra la comunicación ubicua. Pentland comparte con la computación ubicua la idea de convertir a los objetos inanimados que nos rodean, tales como coches, oficinas, ropa... en sensores y actuadores con capacidades de procesamiento. La línea que sigue Pentland ha sido denominada por él mismo como Inteligencia de la Percepción (Perceptual Intelligence). La IP se basa en la capacidad de percepción del dispositivo. Pretende construir dispositivos altamente sensibles que detecten las acciones del usuario y los cambios del entorno, de igual forma que lo haría otra persona. Una vez descubierta la situación actual, los dispositivos podrán actuar en consecuencia adaptándose a las necesidades del usuario. Para Pentland los ordenadores se encuentran muy aislados del mundo exterior, lo que les imposibilita actuar de forma inteligente. La principal diferencia reside en que la IP no pretende crear una red de dispositivos interconectados sino que, al contrario, intenta evitar la comunicación todo lo posible. Los dispositivos tienen capacidades de comunicación, pero ésta sólo se realizará por iniciativa del

usuario. Según Pentland, la Computación Ubica no respeta la intimidad de las personas al facilitar la transmisión de información personal.

En España la investigación en entornos inteligentes tiene sus antecedentes en los trabajos realizados por el grupo CHICO [Ortega, et al 2001]. Este grupo, al frente del cual se encuentra Manuel Ortega, pretende fundir educación y computación ubicua en busca del aula del siglo XXI.

Alcance del presente documento

Acorde con el título del documento, se pretende realizar una revisión de las arquitecturas que permiten el desarrollo de entornos inteligentes. Con lo expuesto en el anterior epígrafe se ha dado una noción del concepto de entorno inteligente. Más complicado sería dilucidar qué se entiende por arquitectura, si no fuera porque en el año 2000 el organismo internacional IEEE (Institute of Electrical and Electronics Engineers) ha propuesto el estándar IEEE 1471, el cual trata de establecer consenso en la forma de describir una arquitectura. Según el estándar [Maier, M. W. et al 2001] se define una arquitectura como “la organización fundamental de un sistema según sus componentes, sus relaciones entre ellos y el entorno, y los principios que guían su diseño y evolución”.

La organización fundamental que se ha elegido para un entorno inteligente consta de tres capas:

- **Capa física:** es la más cercana al mundo real. Se encarga de capturar información del entorno a través de los sensores, y de transmitir las respuestas al usuario utilizando dispositivos audio-visuales y otros actuadores. También es función de esta capa distribuir la información, tanto de entrada como de salida, entre los distintos componentes del entorno inteligente. La capa física está integrada por una o más de redes de datos y un conjunto de dispositivos conectados a estas redes.
- **Capa intermedia:** esta capa es ya totalmente software. Su misión es presentar una visión uniforme a la capa de aplicación de todos los componentes de la capa física. Por regla general la capa física está compuesta por más de una red. Al tener que distribuir información de distinta naturaleza (información proveniente de los sensores, audio, vídeo...), no existe una única red capaz de dar un rendimiento óptimo en todas las situaciones. La capa intermedia no solamente encapsula la complejidad de los distintos dispositivos, sino que también añade información contextual, que aporta un mayor nivel de abstracción al incluir entidades y relaciones que no aparecen directamente en la capa física. Finalmente, además, incluye nuevas funcionalidades a las ofrecidas por la capa física.
- **Capa de aplicación:** está compuesta por módulos, procesos, agentes... que utilizan los servicios ofrecidos por la capa intermedia para realizar aplicaciones específicas del entorno. Esta capa también se encuentra estrechamente relacionada con la interfaz de usuario.

En este documento se van a tratar con detenimiento las dos primeras capas, presentando las distintas alternativas que se pueden encontrar tanto en el ámbito comercial como de investigación. La última capa ha sido descrita en detalle en [Montoro, G. 2000].

Se ha optado por un enfoque centrado en la vivienda, ya que una habitación, o toda la casa entera, son casos particulares donde se puede implementar un entorno inteligente. Se ha tenido en cuenta a la hora de tomar esta decisión la enorme repercusión que las tecnologías de la información están teniendo en la sociedad actual.

Estructuración del presente documento

El resto de este documento se ha dividido en los siguientes capítulos:

- **Capítulo 2. Capa física:** este capítulo se centra en las distintas redes de datos que pueden cohabitar en un entorno inteligente. Se ha separado en tres grandes apartados: redes de control, redes multimedia y redes inalámbricas. Las dos primeras tienen sentido porque el tipo de tráfico que cursan es distinto. Las primeras son redes orientadas a la automatización de la vivienda, lo cual no requiere una elevada velocidad de transmisión. Las segundas cursan audio y vídeo en tiempo real, lo que exige a la red una serie de prestaciones especiales. Las redes inalámbricas se pueden utilizar tanto para lo primero como para lo segundo, pero se han tratado de forma separada porque aportan dos ventajas frente al cable: movilidad de dispositivos y flexibilidad de configuración.
- **Capítulo 3. Capa Intermedia:** se hace un repaso a las diferentes capas intermedias existentes tanto a escala comercial como de investigación. Estos son los dos grandes apartados del capítulo. En ambos se han tratado tanto arquitecturas genéricas como centradas en la vivienda.
- **Capítulo 4.** Se discute InterAct, una arquitectura de pizarra para la implementación de entornos inteligentes. Se presenta la propuesta que se está desarrollando en el Departamento de Ingeniería Informática de la Universidad Autónoma de Madrid tanto para la capa física como para la capa intermedia.
- **Capítulo 5. Conclusiones y trabajos futuros:** Se cierra el documento con una discusión de sus puntos más destacados, y se proponen nuevas líneas de investigación.

Capa física

Dentro de las tecnologías de la información se han aportado diversas soluciones para resolver el transporte y distribución de información. Una primera clasificación, en función de la naturaleza de la información transmitida, separa los sistemas de comunicación en analógicos y digitales. El ámbito de este trabajo se va a centrar exclusivamente en sistemas de comunicación digitales. Estos sistemas han tenido el monopolio de lo que se ha denominado comunicación de datos, esto es, de la comunicación en las redes de ordenadores. Por su parte, los sistemas analógicos históricamente han acaparado los medios de difusión de voz, vídeo y audio, léase telefonía, televisión y radio. Pero el avance en las comunicaciones digitales y las ventajas que estos sistemas ofrecen frente a sus hermanos analógicos han permitido ampliar la definición de lo que se entiende por datos. Esto ha ocasionado que las nuevas redes digitales se diseñen con vista a poder soportar el tráfico multimedia.

Un clasificación tradicional [Tanenbaum, A. K. 1996] de las redes de ordenadores se basa en su extensión. Así habitualmente se han catalogado como:

- WAN (Wide Area Network): su extensión abarca desde un país hasta todo el mundo.
- MAN (Metropolitan Area Network): redes que se extienden dentro de una ciudad.
- LAN (Local Area Network): cuando el recinto donde se encuentran ubicadas abarca centenares de metros.

Recientemente se han acuñado dos nuevos términos que no responden tanto a su extensión sino más bien al emplazamiento en el cual se van a desplegar, o si se prefiere, a la funcionalidad a la cual se van a destinar:

- SOHO (Small Office/ Home Office): que en castellano se conoce como oficina virtual. Son redes orientadas a pequeñas oficinas, como pueden ser las instaladas dentro de una vivienda.
- Home Networking: término que aun no tiene una traducción asentada, y que se refiere al conjunto de todas las infraestructuras digitales que se despliegan en un hogar para comunicar los dispositivos y electrodomésticos entre sí y con el exterior.

Este segundo capítulo se centra en lo que se ha definido como Home Networking, constituyendo un repaso a las distintas redes de comunicación de datos que se pueden encontrar en un hogar. Atendiendo a diversos criterios podemos establecer distintas clasificaciones. La primera división, que es la que se va a seguir como eje del capítulo, se realiza en función del tipo de tráfico que transportan. En este caso se pueden considerar dos categorías de redes:

- Redes de Control: redes cuya filosofía de diseño proviene del campo industrial, y que están enfocadas a tráfico que requiere una baja velocidad de transmisión (<9600 bps).
- Redes Multimedia: redes orientadas a transmitir tráfico que requiere altas velocidades de transmisión (>2 Mbps) y en tiempo real.

Otra distinción importante se establece en función de la licencia que tengan las especificaciones del estándar:

- Abierta: cualquier empresa puede desarrollar productos que utilicen el estándar sin tener que pagar ningún tipo de licencia.
- Propietaria: dependiendo del tipo de licencia se debe pagar por los derechos de utilizar el estándar, o por cada dispositivo fabricado que implemente el protocolo.
- Consorcio: son aquellos estándares que vienen avalados por un conjunto de compañías, que suelen ser líderes en el sector. El estándar es gratuito para las empresas pertenecientes al consorcio.

La política del organismo o empresa que tenga los derechos del estándar puede ser clave en cuanto a la repercusión que pueda tener a escala mundial. Si se trata de un organismo público, o sin ánimo de lucro, el estándar se deja abierto. Si es una empresa, por norma general cobra por el estándar. Posteriormente, el estándar puede que se haga abierto, ya sea porque se considera que se han amortizado las inversiones realizadas, o porque el estándar ha tenido un cierto éxito en el mercado y se quiere conseguir una difusión total.

Por último, se puede establecer una tercera división [Dhir, A. 2001] en función del tipo de cableado de la red:

- No new wires: se aprovechan las instalaciones existentes en la casa.
- New wires: se tiene que introducir un nuevo cableado para desplegar la red.
- Wireless: el estándar se basa en el empleo tecnología inalámbrica. En general la mayoría de los estándares de cable permiten el medio radioeléctrico, pero en este caso no aportan las funcionalidades que promueven los estándares inalámbricos sino que meramente sustituyen el cable por el espacio.

Siguiendo esta clasificación los protocolos descritos en el resto del capítulo quedarían divididos según la Tabla 1:

No new wires	X10, CeBUS, HomePNA
New wires	EIB, LonWork, Nudan, BACNet
Wireless	IEEE 802.11, HomeRF, BlueTooth

Tabla 1. Clasificación en función del cableado de la red

2.1 Redes de datos

2.1.1 Redes de control

La primera cuestión que se plantea, es por qué surge la necesidad de una nueva tecnología distinta de la utilizada en las redes de área local. Hay varios motivos:

- La red de control está pensada para transmitir mensajes cortos.
- Efectúa transmisiones rápidas (ya que las cabeceras son pequeñas).
- Tiene menores costes de implementación.

Se pueden establecer diferentes clasificaciones a las redes de control. Aquí se incluyen cuatro bastante interrelacionadas:

Según su arquitectura

- Bus: todos los dispositivos están conectados de forma lógica a un bus. Habitualmente esta conexión lógica coincide con un mismo medio físico. El algoritmo más utilizado para resolver el acceso al medio es el CSMA/CD² (véase 2.1.3) o variantes propietarias.
- Estrella: existe un dispositivo central al cual se conectan el resto de los componentes de la red.
- Anillo: los dispositivos se conectan uno a uno formando un círculo.
- Ad hoc (o topología libre): los dispositivos se suelen conectar uno a uno pero sin formar una red regular distinguible.

² CSMA/CD: Carrier Sense Multiple Access/ Collision Detect

Según el medio físico:

- Cableados: ya sea par trenzado (categoría 3 o 5), coaxial o fibra óptica. Tienen la ventaja de que se consigue una comunicación fiable ya que la señal se encuentra confinada dentro del cable.
- Inalámbricos: se basan en interfaces radio. Permiten configurar y dismantelar una red de forma sencilla y barata. En un futuro se piensa en que los terminales sean capaces de detectar y comunicarse con otros terminales sin orden explícita del usuario.

Según el control:

- Centralizado: un dispositivo es el que posee toda la inteligencia del sistema. Las comunicaciones siguen el esquema maestro/esclavo.
- Distribuido: todos los dispositivos conectados tienen una cierta autonomía. Las comunicaciones son entidad-a-entidad.

Según el modelo de comunicación:

- Orientado a datos: los dispositivos actúan como si compartieran una memoria global. Un dispositivo no interacciona directamente sobre otro, sino que cambia alguna variable y el otro dispositivo al observar el cambio obra en consecuencia.
- Orientado a comandos: un dispositivo puede ejecutar una acción directa sobre otro dispositivo.

En un principio las redes de control se basaron en protocolos con topología de estrella, control centralizado y orientados a comandos. La nueva tendencia es utilizar topología de bus, control distribuido y orientado a datos.

En la Tabla 2 se muestra una clasificación de los protocolos estudiados según los tres criterios antes expuestos:

Protocolo/Red	Topología	Control	Modelo
EIB	Bus	Distribuido	Datos
X10	Bus	Distribuido	Comandos
LonWork	Bus	Distribuido	Datos
CeBus	Bus	Distribuido	Comandos
NUDAN	Bus	Distribuido	Datos
BatiBus	Estrella	Centralizado	Comandos
BACNet	Bus	Distribuido	Comandos

Tabla 2. Clasificación de diferentes protocolos de control en función de topología, control y modelo de comunicación.

A continuación se discuten cada uno de estos protocolos:

2.1.1.1 EIB (European Installation Bus)

El European Installation Bus [Goossens M. 1998] (EIB a partir de ahora) se ha pensado para ser utilizado como un sistema de gestión para la instalación eléctrica de un edificio. Su propósito comprende la monitorización y control de sistemas que requieren tráfico

con baja velocidad de transmisión, tales como el alumbrado, la calefacción, el aire acondicionado, ventilación, persianas y alarmas de un edificio.

El estándar EIB ha sido propuesto por la EIBA (European Installation Bus Association). La EIBA es una organización que reúne a empresas europeas, al frente de la cual se encuentra la empresa Siemens. Pertenecen a la asociación más de 100 miembros que como fabricantes cubren el 80% de la demanda de aparatos de instalación eléctrica en Europa. Esto asegura que los usuarios continuaran disponiendo de la libre elección de los productos de diferentes fabricantes y de sus soluciones técnicas. Aunque próximamente será estándar ANSI, la penetración en EE.UU. es muy baja.

Tecnología

El bus EIB se puede definir como un sistema descentralizado en el que cada uno de los dispositivos conectados tiene control propio, mediante su propio microprocesador. Estos dispositivos se pueden clasificar en sensores, que son los responsables de detectar actividad en el edificio, y en actuadores, que son capaces de modificar el entorno.

La EIBA propone una especificación abierta en la cual todos los dispositivos se conectan a través de una única línea de bus. Para solucionar el acceso al medio físico compartido se utiliza el protocolo CSMA/CA³. Los sensores se comunican mandando telegramas a los actuadores, los cuales ejecutan los comandos apropiados.

El bus se adapta fácilmente a distintos tamaños y topologías pudiéndose conectar del orden de 10.000 dispositivos. Es independiente del medio físico que se utilice, estando disponible para los siguientes:

- Par trenzado (9600bps).
- Red eléctrica (1200/2400bps, en un principio para 230V y 50Hz)
- EIB.net (10 Mbps sobre Ethernet)
- Radiofrecuencia.
- Infrarrojos.

Las instalaciones existentes en la actualidad están implementadas sobre par trenzado y en menor medida sobre red eléctrica, pudiendo tener elementos que se comunican mediante infrarrojos o radiofrecuencia.

Topología

Tal como se muestra en la Figura 1, la red del EIB se estructura de forma jerárquica. La unidad más pequeña se denomina línea, a la cual se pueden conectar hasta un máximo de 64 dispositivos. Las líneas se agrupan en áreas. Cada área se compone de una línea principal de la cual cuelgan hasta 15 líneas secundarias. Por tanto, un área podrá tener como máximo 960 dispositivos. Cada una de las líneas secundarias se conecta con la línea principal mediante un dispositivo llamado acoplador de línea. Cada línea principal deberá tener su propia fuente de alimentación.

A su vez se puede disponer de hasta 15 áreas unidas mediante una línea principal denominada *backbone*. Como máximo se podrán conseguir hasta 14.400 dispositivos. Las áreas se conectan al *backbone* mediante acopladores.

³ CSMA/CA: Carrier Sense Multiple Access with Collision Avoidance.

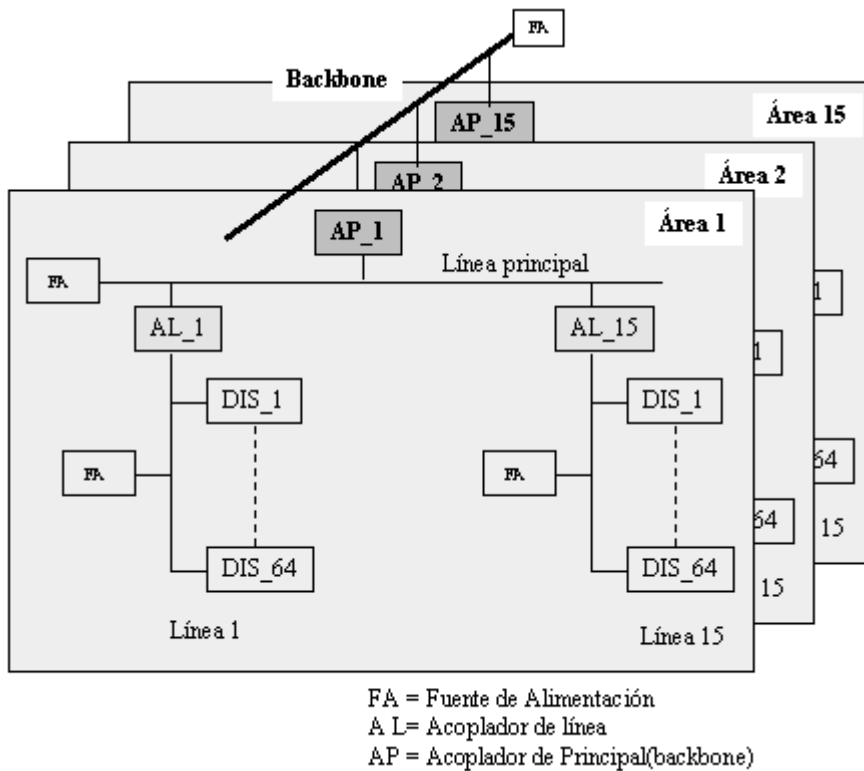


Figura 1. Topología de una red EIB

Para identificar unívocamente los dispositivos en la red, cada uno tiene asociada una dirección física de 16 bits. La dirección de un dispositivo además define la localización de éste en la red, ya que cada dirección indica el área, la línea dentro del área, y el número dentro de la línea para cada dispositivo (véase Figura 2).

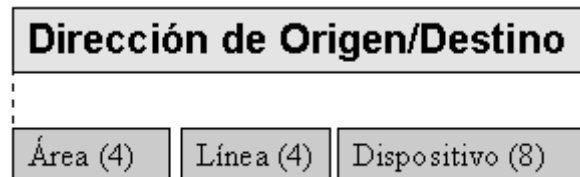
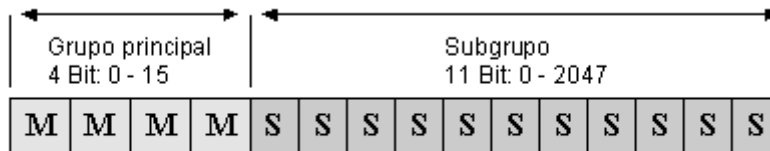


Figura 2. Formato de una dirección física EIB

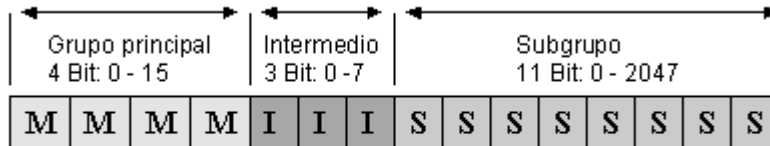
Además de la dirección física, cada dispositivo puede tener una o más direcciones lógicas, denominadas direcciones de grupo. Las direcciones de grupo asocian funcionalmente dispositivos. De este modo, todos los dispositivos que tengan la misma dirección de grupo reciben los mismos mensajes. Los sensores sólo pueden enviar telegramas a una dirección de grupo, mientras que los actuadores pueden tener varias direcciones de grupo, lo que les permite reaccionar a distintos sensores. Cualquier dispositivo de la red puede mandar telegramas a una dirección de grupo.

Dependiendo de la granularidad que el diseñador quiera dar a la red se pueden seleccionar direcciones de grupo de nivel 2 o de nivel 3. Las direcciones de grupo de nivel 2 (véase Figura 3) dividen la dirección en dos campos: grupo principal y subgrupo. Por otro lado, las de nivel 3 separan la dirección en: grupo principal, grupo intermedio y subgrupo (véase Figura 4). Con el nivel 2 se obtienen 15 grupos principales con 2047 subgrupos cada grupo. Para el nivel 3 la división queda en 15 grupos principales, cada uno con 7 grupos intermedios de 255 subgrupos cada uno.



Dirección de Grupo: Nivel 2

Figura 3. Formato de una dirección de grupo EIB de nivel 2



Dirección de Grupo: Nivel 3

Figura 4. Formato de una dirección de grupo EIB de nivel 3

Telegramas

Los dispositivos se comunican mediante señales binarias en banda base con una velocidad de transmisión de 9600 bps (en el caso de cable trenzado). Un cero lógico se representa mediante un pulso de corriente mientras que la ausencia de ésta significa un uno lógico (véase Figura 5).

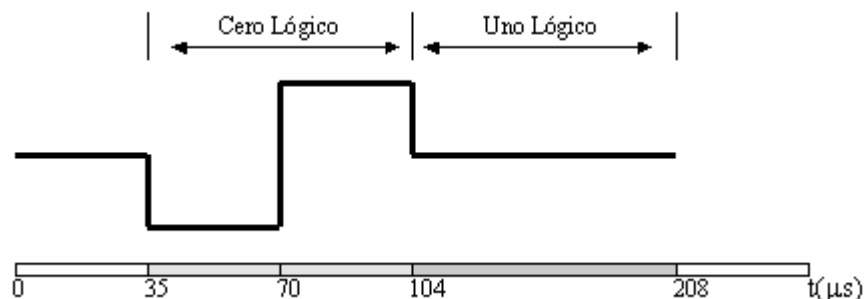


Figura 5. Pulso de transmisión EIB

Al tener que compartir el medio físico de transmisión, un dispositivo comenzará a transmitir siempre y cuando el bus éste desocupado. Cuando dos o más dispositivos transmiten simultáneamente se produce una colisión en el bus que será resuelta mediante un algoritmo CSMA/CD. Los dispositivos se mantienen a la escucha mientras están transmitiendo. Tan pronto como detecten un cero cuando ellos estén transmitiendo un uno, se pararán dejando el bus libre para el dispositivo de mayor prioridad.

El intercambio de información entre dos dispositivos se consigue mediante el envío de telegramas. Un telegrama se compone de un paquete de datos estructurado que el emisor envía, y del correspondiente acuse de recibo con el que el receptor responde si no ha ocurrido ningún fallo. Cada paquete de datos se divide en los siguientes campos:

- Control (8 bits).
- Dirección del emisor (16 bits).
- Dirección del destinatario (16 bits +1 bit).
- Contador (3 bits).
- Longitud (4 bits).
- LSDU (Link Service Data Unit): que es la información que se transmite (hasta 16x8 bits).
- Byte de comprobación (8 bits).

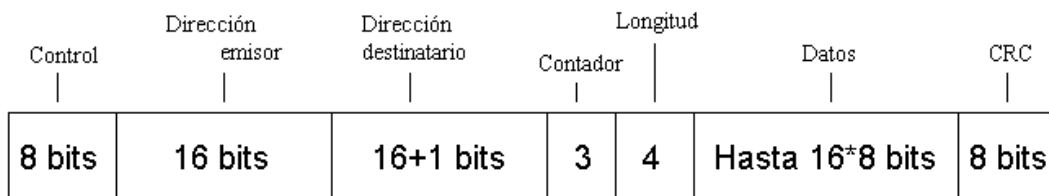


Figura 6. Formato de un telegrama EIB

El campo de control sirve para determinar la prioridad del mensaje, así como para marcar el inicio del telegrama.

Tanto la dirección del emisor como la del receptor siguen el formato explicado anteriormente, añadiendo un bit más en la dirección del destinatario que indica si se trata de una dirección física o de una dirección de grupo.

El contador se utiliza para funciones de encaminamiento, contando el número de saltos que ha dado el paquete. La longitud indica cuantos bytes ocupa la LSDU.

El último byte se utiliza para comprobar que los anteriores han sido transmitidos correctamente.

Semántica de datos

Para que dos dispositivos puedan comunicarse no sólo precisan conocer cómo localizarse entre sí sino que también deben compartir una semántica común. Los datos intercambiados tienen que tener el mismo significado para los dos dispositivos. EIB soluciona este problema definiendo el estándar EIS⁴.

EIS define hasta catorce tipos de datos distintos. Cada tipo de datos se caracteriza por el número de bits que ocupa y el significado de los distintos valores que pueden tomar. Por ejemplo, el tipo de datos más simple es el EIS 1, que ocupa 1 bit y se utiliza para funciones de conmutación tales como encender/apagar, habilitar/deshabilitar, alarma/no alarma, verdadero/falso, etc.

⁴ EIB Interworking Standard

Dispositivos

Cada dispositivo que se conecta al bus esta organizado en tres módulos:

- Unidad de acoplamiento al bus (BCU - Bus Coupling Unit)
- Módulo de la aplicación (AM - Application Module)
- Programa de la aplicación (AP - Application Program)

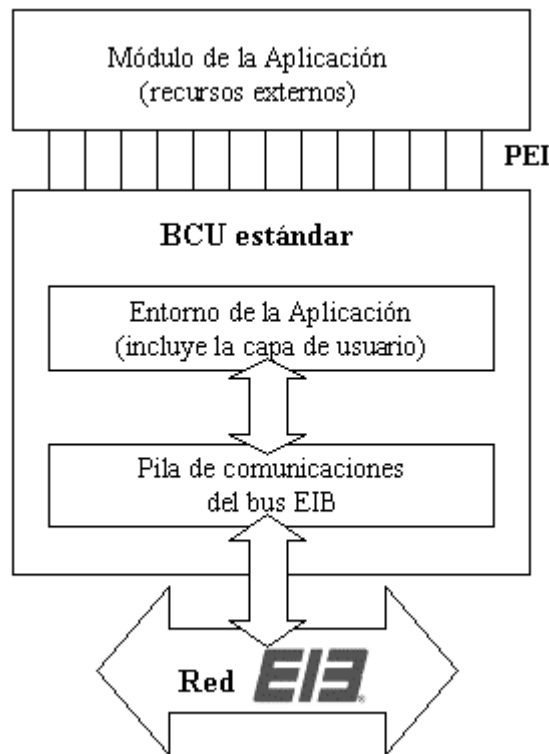


Figura 7. Arquitectura de un dispositivo EIB

Aunque existen dispositivos que integran las tres partes sin que se perciba una diferencia clara, las especificaciones están pensadas para que cada módulo pueda adquirirse de un fabricante distinto. Para el caso de que el módulo BCU y el módulo AM se adquieran por separado, se ha normalizado la conexión entre ambos mediante lo que se denomina la Interfaz Externa y Física (PEI Physical External Interface). Ésta es una interfaz a nivel hardware que define hasta catorce modos de comunicación.

El módulo BCU se ocupa de implementar la parte de comunicación con el bus, así como de mantener el estado interno del dispositivo. Se puede dividir en dos partes: el transmisor y el controlador. El controlador consiste en un microprocesador con un mapa de memoria formado por una ROM, una RAM y una EEPROM.

- La memoria ROM contiene una serie de programas del sistema grabados durante la fabricación que se encargan de implementar la pila de comunicación del bus EIB y el comportamiento genérico del módulo BCU.
- La RAM se utiliza como puente de comunicación entre los programas del sistema y el programa de la aplicación.
- La EEPROM almacena tanto parámetros del sistema como el programa de la aplicación.

La parte del sistema más importante del BCU son los objetos de comunicación. Estos contienen información relevante sobre el estado del dispositivo, por ejemplo, si una lámpara está encendida o apagada, la hora y fecha de un reloj, o si se ha pulsado un determinado interruptor. Cada dispositivo puede tener uno o más objetos de comunicación. El tipo de datos de los objetos de comunicación tiene que ser alguno de los definidos en el estándar EIS. Cada objeto de comunicación tiene una dirección de grupo asociada. Ésta es única si se trata de un objeto de comunicación emisor o múltiple si es un objeto de comunicación receptor. Un objeto de comunicación emisor y otro receptor se ligan entre sí mediante la asociación de una misma dirección de grupo, siempre y cuando ambos objetos sean del mismo tipo. Cuando cambia el valor del emisor, la BCU se encarga de transmitir el nuevo valor al grupo asociado. Todos los objetos de comunicación receptores que tengan la misma dirección de grupo se enterarán del cambio y actuarán en consecuencia.

El módulo AM se encarga de particularizar cada aplicación en concreto. Por ejemplo, si se trata de un interruptor, el módulo AM tendrá que generar una señal eléctrica cada vez que se oprima el pulsador. El BCU detectará el cambio de estado en el pulsador gracias a la interfaz PEI. Actualmente sólo se disponen módulos BCU para los medios físicos de par trenzado y de red eléctrica. Se están desarrollando módulos BCU para radiofrecuencia e infrarrojos.

En el caso de los sensores, el módulo de la aplicación (AM) transferirá la información que recoge del entorno a la BCU a través de la PEI. La BCU codificará y enviará los datos recogidos a través del bus. La BCU será la encargada de comprobar periódicamente el estado del sensor. Para los actuadores, se realiza el proceso inverso, la BCU recibe los telegramas del bus, los decodifica y pasa la información al AM.

Mientras que los dos primeros son módulos hardware, el programa de la aplicación engloba toda la parte software del dispositivo, que será diferente para cada uno según la función que realice. Cambiando el programa de la aplicación se puede modificar rápidamente el comportamiento de un dispositivo sin tener que modificar los componentes físicos.

Conclusiones

El bus EIB se perfila como uno de los estándares que puede dominar el mercado en los próximos años. Dentro del mercado europeo se encuentra muy bien posicionado ya que está respaldado por las grandes empresas del sector. Recientemente ha surgido una iniciativa denominada Konnex (véase 2.1.1.6) que establece criterios de convergencia entre los tres estándares más importantes de Europa: EIB, EHS y Batibus. Esta iniciativa toma como piedra angular a EIB. Tras esta fusión, EIB se queda sin competidores, aunque el estándar americano LONWork (véase 2.1.1.3) está comenzado a posicionarse en Europa. Es complicado pronosticar cuál de los dos estándares sobrevivirá, o si convivirán pacíficamente. Si LONWork acaba por imponerse en el mercado americano, se podría desequilibrar la balanza en el mercado europeo.

Dentro del campo de la investigación, se están realizando progresos en el desarrollo de pasarelas con TCP/IP [Reiter, H. 1998], integración con Jini [Kastner W., y Krügel C. 2000], y agentes inteligentes [Palensky, P. 2000]. En España, EIB se ha aplicado tanto en edificios privados como en públicos, siendo sin embargo escasos los proyectos de investigación relacionados [Fernández-Valdivieso, C. et al 1999] [Fernández-Valdivieso, C. et al. 2000].

2.1.1.2 X10

X10 [x10] es un protocolo de comunicación que permite controlar aparatos eléctricos a través de la instalación de red eléctrica. El estándar surgió hace 20 años como parte de los experimentos realizados por la empresa Picosystem y lleva más de quince funcionando en el ámbito comercial.

La transmisión de una señal binaria en X10 se realiza mediante ráfagas de 120 kHz superpuestas en los cruces por cero de la señal de la red eléctrica. La ausencia de ráfaga significa un cero lógico.

En un principio, se distinguieron dos tipos de dispositivos X10, los transmisores (transmitters) y los receptores (receivers). Los transmisores envían comandos X10 codificados como una señal de baja intensidad que se superpone a la señal de la red. Se pueden enviar mensajes a un máximo de 256 dispositivos en una misma red. Cada uno de los receptores tiene una dirección. Éstos son capaces de demodular la señal y si corresponde con su dirección actuar en consecuencia. Varios receptores pueden tener la misma dirección de tal forma que se puede actuar sobre ellos a la vez. Como los receptores no son capaces de responder, es imposible saber su estado. Por ejemplo no se puede preguntar si una luz está o no encendida. Para solucionar este problema se introdujo un tercer tipo de dispositivos que permitía transmitir y recibir. Aun así siguen siendo muy comunes en las instalaciones los dispositivos con una sola función.

En la trama del protocolo se incluye un campo delimitador (start code), la dirección del dispositivo (compuesta del house code y el device code), y la función o comando X10 que se quiere realizar.

Conclusiones

X10 es un protocolo muy extendido en Estados Unidos sobre el cual se han realizado una enorme cantidad de desarrollos. La mayoría de las instalaciones domóticas residenciales utilizan este protocolo. Incluso en la actualidad se puede encontrar como bus de control en proyectos de investigación [Coen, M. H. 1998]. Las ventajas que han llevado a X10 a ser tan popular son:

- Es sencillo de utilizar.
- Es fácil de implementar y los dispositivos son baratos.
- No hace falta desplegar una nueva red, sino que se utilizan las infraestructuras existentes.

Por otro lado, en Europa este protocolo no ha tenido casi ninguna repercusión, principalmente debido a las diferencias en tensión entre el estándar americano y europeo de red eléctrica⁵. Esto conlleva a un encarecimiento de los precios para los productos que se importan de Estados Unidos, que ha frenado considerablemente la expansión de este protocolo en Europa.

⁵ En EEUU la tensión trifásica es de 110 V mientras que en Europa es de 220V. También existen diferencias en la frecuencia que es de 60 Hz y 50 Hz respectivamente.

2.1.1.3 LONWORK

LONWork [LONWork 1999] es un estándar propietario desarrollado por la empresa Echelon. El estándar ha sido ratificado por la organización ANSI como oficial en Octubre de 1999 (ANSI/EIA 709.1-A-1999).

El estándar LONWork se basa en el esquema LON (Local Operating Network). Éste consiste en un conjunto de dispositivos inteligentes, o nodos, que se conectan mediante uno o más medios físicos y que se comunican utilizando un protocolo común. Por inteligente se entiende que cada nodo es autónomo y proactivo. Esto posibilita que pueda ser programado para enviar mensajes a cualquier otro nodo como resultado de cumplirse ciertas condiciones, o para llevar a cabo ciertas acciones en respuesta a los mensajes recibidos.

El funcionamiento completo de la red surge de las distintas interconexiones entre cada uno de los nodos. Mientras que la función desarrollada por uno de los nodos puede ser muy simple, la interacción entre todos puede dar lugar a aplicaciones complejas. Uno de los beneficios inmediatos de LON es que un pequeño número de nodos puede realizar un gran número de distintas funciones dependiendo de cómo estén interconectados.

LONWork utiliza para el intercambio de información (ya sea de control o de estado) el protocolo LonTalk. Toda la información del protocolo está disponible para cualquier fabricante, aunque Echelon exige una cierta tasa por cada dispositivo fabricado.

Protocolo LonTalk

LonTalk ha sido creado dentro del marco del control industrial por lo que se enfoca a funciones de monitorización y control de dispositivos. Dentro de este marco se han potenciado una serie de características:

- **Fiabilidad:** El protocolo soporta acuse de recibo (acknowledgments) extremo a extremo con reintentos automáticos.
- **Variedad de medios de comunicación:** tanto cableado como por radio. Entre los que se soportan: Par trenzado, red eléctrica, radiofrecuencia, cable coaxial y fibra óptica.
- **Tiempo de Respuesta:** Se utiliza un algoritmo propietario para la predicción de colisiones que consigue evitar la degradación de prestaciones que se produce por tener un medio de acceso compartido.
- **Bajo coste de los productos:** Muchos de los nodos LON son simples dispositivos como interruptores o sensores. El protocolo ha sido diseñado para poder ser implementado en un único chip de bajo coste.

Para simplificar el encaminamiento de mensajes, el protocolo define una jerarquía de direccionamiento que incluye dirección de dominio, subred y nodo. Cada nodo está conectado físicamente a un canal. Un dominio es una colección lógica de nodos que pertenecen a uno o más canales. Una subred es una colección lógica de hasta 127 nodos dentro de un dominio. Se pueden definir hasta 255 subredes dentro de un único dominio. Todos los nodos de una subred deben pertenecer al mismo canal, o los canales tienen que estar conectados por puentes (bridges). Cada nodo tiene un identificador de 48-bits único, asignado durante la fabricación, que se usa como dirección de red durante la instalación y configuración. La Tabla 3 resume la jerarquía de red:

Subredes por dominio:	255
Nodos por subred:	127
Nodos por dominio:	32.385
Grupos por dominio:	255
Nodos por grupo:	63
Numero de dominios:	281.474.976.710.656

Tabla 3. Número de dispositivos LONWork permitido de forma desglosada

LonTalk es un estándar abierto que puede ser implementado por cualquier fabricante de circuitos integrados. El circuito integrado más difundido que implementa el protocolo se denomina *Neuron* y lo fabrican las empresas Cypress, Toshiba y Motorola.

Variables de Red (Network Variables):

La comunicación entre nodos se completa con las variables de red. Cada nodo define una serie de variables de red que pueden ser compartidas por los demás nodos. Cada nodo tiene variables de entrada y de salida, que son definidas por el desarrollador.

Siempre que el programa que se ejecuta en un nodo escribe un nuevo valor en una de sus variables de salida, éste se propaga a través de la red a todos los nodos cuyas variables de entrada estén conectadas a esta variable de salida. Todas estas acciones están implementadas dentro del protocolo. Sólo se podrán ligar variables de red que sean del mismo tipo.

Para guardar la interoperabilidad entre productos de distintos fabricantes, se especifican las variables a partir de una definición de tipos estándar (Standard Network Variable Types). Echelon mantiene una lista de unos 100 tipos a disposición de cualquier fabricante.

Componentes de una red LONWork:

Se pueden distinguir dos partes:

- **Transmisor LONWork (Transceivers):** Estos dispositivos sirven de interfaz entre el chip *Neuron* y el medio físico. Dependiendo del medio físico la velocidad de transmisión y topología es distinta
- **Circuito Integrado *Neuron*:** Es el corazón de la tecnología LONWork. Contiene toda implementación del protocolo LonTalk. Cada CI *Neuron* tiene tres procesadores de 8-bit, dos dedicados al protocolo y un tercero a la aplicación del nodo.

Conclusiones

LONWork es un protocolo muy similar al bus EIB, ya que su filosofía de diseño es prácticamente la misma. A nivel técnico se puede considerar que tiene las mismas ventajas y desventajas que EIB, salvando las diferencias entre los circuitos integrados que implementan los respectivos protocolos. Por otro lado LONWork presenta mayor difusión en Europa de la que tiene EIB en EE.UU. Tanto en el campo investigador como el campo comercial se pueden encontrar desarrollos realizados con este protocolo, especial mención en este último campo al Centre of Excellence for Fieldbus Systems de la Universidad Tecnológica de Viena.

2.1.1.4 CEBus

El protocolo de comunicación CEBus [Douligeris, C. 1993] [CEBus 1995] (Consumer Electronics Bus) es un estándar vigente en los Estados Unidos que ha sido desarrollado por la Asociación de Industrias Electrónicas (EIA-Electronic Industries Association). El estándar surgió en 1984 cuando la EIA se propuso unificar los protocolos de señalización infrarroja para el control remoto de electrodomésticos. En 1992 el estándar se había extendido a todo el ámbito de control domótico.

Los objetivos principales del estándar son:

- Facilitar el desarrollo de módulos de interfaz de bajo coste que puedan ser integrados fácilmente en electrodomésticos.
- Soportar la distribución de servicios de audio y vídeo tanto en formato analógico como digital.
- Evitar la necesidad de un controlador central, distribuyendo la inteligencia de la red entre todos los dispositivos.
- Permitir añadir y quitar componentes de la red sin que afecte al rendimiento del sistema ni se requiera un gran esfuerzo de configuración por parte del usuario.
- Proporcionar un método adecuado de acceso al medio.

Medios físicos permitidos

- Red eléctrica
- Cable trenzado
- Cable coaxial
- Infrarrojos
- Radiofrecuencia
- Fibra óptica
- Bus audio-vídeo

En todos los medios físicos, la información de control y datos se transmite a la misma velocidad: 8000 bit/s. Sin embargo también se permiten canales para acomodar audio o vídeo.

Funcionamiento

Los comandos y el estado de los dispositivos se transmiten por el canal de control en forma de mensajes. El núcleo de la especificación CEBus se centra en definir este canal de control. El formato de los mensajes CEBus es independiente del medio físico utilizado. Cada mensaje contiene la dirección de destino de receptor sin ninguna referencia al medio físico en el que está situado el receptor o el transmisor. De esta forma CEBus forma una red uniforme a nivel lógico en forma de bus. CEBus soporta una topología flexible. Cualquier dispositivo se puede conectar a cualquier medio siempre que tenga la interfaz adecuada. Para comunicar segmentos de red que tienen diferente medio físico, se utilizan dispositivos llamados encaminadores. Estos pueden estar integrados dentro de otro dispositivo con más funcionalidades.

Para facilitar la difusión de mensajes todos los dispositivos tienen una dirección a la que responden siempre (broadcast address). Además, los dispositivos se pueden agrupar

en grupos (group address), pudiendo un dispositivo pertenecer a uno o más grupos. De esta forma se puede mandar un único mensaje a varios dispositivos al mismo tiempo.

CAL (Commun Appliance Language)

CAL [CAL1 1996] [CAL2 1996] es el lenguaje que utilizan los dispositivos CEBus para comunicarse. Es un lenguaje orientado a comandos que permite controlar dispositivos CEBus y asignar recursos.

El lenguaje define un conjunto de funciones para cada uno de los dispositivos. Hay dos tipos de funciones: aquellas que gestionan la asignación de recursos, tales como pedir, usar y liberar recursos, y las funciones de control, que proporcionan la capacidad de enviar comandos CAL a dispositivos remotos y responder a comandos CAL.

CAL utiliza el paradigma de programación orientada a objetos. Cuando un objeto recibe un mensaje se ejecuta alguno de los métodos disponibles. Un mensaje consiste en un identificador de método seguido opcionalmente de parámetros. Cuando se recibe el mensaje, se busca en la lista de métodos cuál es el que tiene el identificador y si se encuentra, se ejecuta. CAL, por tanto, lo que define es el conjunto de funciones que ofrece un dispositivo.

Los objetos CAL no se organizan en jerarquías (no existe el concepto de herencia tal como se entiende en OO) sino que el comportamiento depende del contexto en el que se encuentre. Por ejemplo, si tenemos un objeto de control analógico, éste se puede usar tanto para representar un control de volumen, un termostato o un regulador. La función exacta vendrá determinada por el contexto en el cual sea instanciado el objeto.

Conclusiones

CEBus es más parecido a X10 en cuanto a la filosofía de diseño, ya que también se basa en un lenguaje orientado a comandos. Sin embargo hay que tener en cuenta que CEBus supera con creces las deficiencias que presenta X10. Mientras que EIB y LONWork son estándares genéricos de control, CEBus ha sido diseñado claramente orientado al control de sistemas domóticos.

Aunque la Asociación de Industrias Electrónicas ha realizado un enorme esfuerzo en fomentar el estándar, hasta el momento ha tenido una baja repercusión. No obstante, la propuesta de Microsoft para redes domóticas (ver 3.1.6), ha elegido como soporte el estándar CEBus. Esto ha originado que CEBus esté disfrutando de una segunda oportunidad después de varios años de existencia con una implantación escasa. De momento es pronto para aventurar el futuro del estándar debido a que su principal apuesta tiene un tiempo de vida todavía corto.

2.1.1.5 NUDAN

NUDAN [Warriner, P. y Karam, K. Z. 1998] es el acrónimo de Newcastle University Domestic Area Network. Es una propuesta del Departamento de Ingeniería Eléctrica y Electrónica de la Universidad de Newcastle. El estándar persigue la idea de poder desarrollar una red multifuncional de bajo coste.

NUDAN se basa en un único bus al cual se acoplan los dispositivos en cualquier punto y en cualquier momento. Gracias a su velocidad de transmisión de hasta 40 Mbps permite transmitir desde simples comandos de control hasta información compleja como ficheros de texto, sonido e imágenes.

NUDAN modela la red como un conjunto de objetos, cada uno con un identificador único de 16 bits. También contiene información sobre su localización, operación y

capacidad de comunicación. No existe un control centralizado sino que se distribuye en toda la red, siendo las comunicaciones entre objetos asíncronos.

La red física es un único bus independiente del medio físico. Para solucionar el problema de acceso al medio se utiliza una variante de CSMA (ver 2.1.3).

Los dispositivos se añaden al bus mediante un módulo intermedio. Este módulo contiene un sistema operativo genérico (GOS) que es el responsable de implementar el protocolo.

2.1.1.6 **Konnex**

Konnex surge como una iniciativa de tres asociaciones europeas:

1. EIBA, (European Installation Bus Association).
2. Batibus Club International.
3. EHS (European Home Systems Association).

con el objeto de crear un único estándar europeo para la automatización de las viviendas y oficinas. Este año ha salido la primera versión de la especificación, la cual ha sido creada a partir de los tres protocolos pertenecientes a cada organización, siendo compatible con los productos EIB ya instalados.

Del bus EIB ya se han comentado ampliamente sus características (véase 2.1.1.1). Los otros dos buses se describen brevemente a continuación:

EHS

El EHS [EHS] fue inicialmente propuesto por la empresa Trialog, y posteriormente adoptado por la EHS (European Home System Association). Ésta es un consorcio abierto que pretende impulsar a la industria Europea dentro del campo de la domótica. Dentro de los miembros del consorcio destacan las empresas: Bosch Telecom, Deutsche Telekom, France Telecom, Siemens, Philips, HoneyWell, Caba, Home Automation Association y el Instituto Tecnológico Aragonés.

Las especificaciones del bus están disponibles para red eléctrica (2400 bps) y par trenzado de baja velocidad (48Kbps), ambas con variantes del CSMA para regular el acceso al medio⁶. La red se entiende como un conjunto de segmentos que comparten el mismo medio y unidos mediante encaminadores. Se tienen tres niveles de direccionamiento (a nivel de enlace, de red y de aplicación). Cada segmento puede tener hasta 256 dispositivos, mientras que todo el sistema permite hasta 10^{12} dispositivos. El control es distribuido y orientado a comandos.

Aunque se han desarrollado varios proyectos ESPRIT utilizando el EHS, las aplicaciones y productos existentes en el mercado son escasos.

Este protocolo tiene prestaciones y características similares al CEBus americano (véase 2.1.1.4).

Batibus

Fue desarrollado por la empresa francesa Merlin. Se basa en la tecnología de par trenzado, pudiendo transmitir hasta 4800 bps. El sistema es centralizado. Cada central controla hasta 500 puntos de red. El estándar se ha quedado obsoleto debido a sus limitaciones, aunque ha tenido cierta repercusión.

⁶ Se prevé acaparar más medios físicos como coaxial, infrarrojo, fibra óptica...

2.1.1.7 BACNet

El BACnet [Newman, H.M. 1997] es un protocolo abierto de comunicación de datos para la automatización de viviendas y redes de control. BACNet ha sido desarrollado bajo el auspicio de la Asociación Americana de Fabricantes de equipos de Calefacción, Refrigeración y Aire Acondicionado, en inglés la Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE).

El objetivo principal de la Asociación fue el de crear un protocolo abierto que permitiera interconectar equipos de distintos fabricantes de forma transparente. BACNet fue una de las primeras tentativas para formalizar la comunicación entre los distintos dispositivos que se pueden encontrar en un hogar, comenzado su andadura a principios de los ochenta.

Las especificaciones de BACNet se dividen en tres partes: la primera se encarga de poder representar cualquier dispositivo utilizando un lenguaje estándar. La segunda define el conjunto de mensajes que se pueden enviar para controlar un dispositivo, y la última especifica el conjunto de configuraciones de red que se pueden dar en una red BACNet.

Conclusiones

BACNet, al igual que CEBus, es un protocolo claramente orientado al control domótico. La parte más interesante de este protocolo es el esfuerzo que se ha realizado para definir un conjunto de reglas hardware y software que permita comunicarse a dos dispositivos independientemente de si estos utilizan protocolos como el EIB, el BatiBUS, el EHS, el LonTalk, TCP/IP, etc. Gracias a que es uno de los primeros estándares que surgió, y gracias también al respaldo de la ASHRAE, BACNet tiene asegurada su permanencia en el mercado, aunque hasta el momento restringida al ámbito norteamericano.

2.1.2 Redes Multimedia

Con la digitalización del vídeo y del audio han surgido nuevos protocolos que pretenden solucionar la interconexión de los nuevos aparatos electrónicos, tanto entre sí como con los ordenadores personales. Este esfuerzo en transmitir todas las señales con formato digital permite reducir los costes de interconexión y homogeneizar los equipos.

Se ha utilizado la denominación de redes multimedia para designar a aquellas redes que son capaces de cursar voz y vídeo en tiempo real. Este tipo de tráfico tiene dos requerimientos que imponen unas características mínimas a las redes multimedia.

La primera restricción es que el tráfico multimedia necesita un ancho de banda mínimo para poder transmitirse. Éste además es bastante elevado, especialmente en el caso del vídeo, y aumenta cuanto mayor sea la calidad exigida. En la Tabla 4 se muestran tanto para audio como para vídeo, los diferentes anchos de bandas mínimos requeridos para distinta calidades.

Contenido	Ancho de Banda requerido
Audio	
• Calidad CD	706 Kbit/s (44100 muestras/s, 16-bit x muestra)
• Calidad telefonía digital	64 Kbit/s (8.000 muestras/s, 8-bit x muestra)
Vídeo	
• Calidad mínima	566 Kbit/s (1024x768, 30 marcos/s, 3 color, 8-bit x color)
• Calidad TV	
○ Sin comprimir	96 Mbit/s
○ HDTV	19 Mbit/s
○ DVD	3-8 Mbit/s
○ MPEG-1	1.5 Mbit/s
○ MPEG-2	6 Mbit/s

Tabla 4. Clasificación de distintos tipos de tráfico multimedia según el ancho de banda requerido

Tal como se muestra a lo largo del apartado 2.1.1, los protocolos de control no tienen el ancho de banda suficiente como para poder cursar tráfico multimedia. Por otro lado, la infraestructura de las redes que se explican en este apartado es demasiado cara como para que sean utilizadas también como redes de control. Esto da lugar a que, por el momento, existan dos redes en paralelo dentro de la vivienda automatizada.

La segunda restricción viene impuesta porque el tráfico multimedia es isócrono. Cuando se transmite la señal de vídeo/audio digitalizada, se están enviando en cada paquete muestras de la señal. El receptor según le llegan los paquetes extrae las muestras y puede reconstruir la señal analógica. Para que la reconstrucción sea en tiempo real, el retardo entre dos paquetes consecutivos no puede superar un máximo. Esto obliga a que la red disponga de mecanismo que asegure para toda la transmisión un retardo aceptable.

En los siguientes apartados se presentan distintos protocolos que soportan tráfico multimedia:

2.1.2.1 IEEE 1394 (FireWire)

Es un bus de comunicaciones desarrollado por Apple [IEEE 1394 1995], pensado para interconectar aparatos electrónicos digitales de consumo, tales como videocámaras, reproductores de DVD, televisores, impresoras, electrodomésticos y, por supuesto, ordenadores personales.

El bus permite soportar velocidades de hasta 400 Mbps (en un futuro se está pensando en ampliar a 1GHz), garantizando un ancho de banda suficiente para aplicaciones multimedia. Los dispositivos se conectan siguiendo la filosofía Plug and Play, permitiendo conectarlos con el bus funcionando. El bus se organiza como si existiera un espacio de memoria compartido por todos los dispositivos. Cada dispositivo se direcciona mediante 64 bits, repartidos en 10 bits para el identificador de red, 6 bits para el identificador del nodo y 48 bits para el direccionamiento de memoria. El resultado es poder direccionar 1023 redes de 63 nodos cada una y con 281 Terabytes de memoria. Este direccionamiento de memoria en vez de estar basado en canales permite ver los recursos como si fueran registros.

Cada entidad del bus se denomina nodo. Cada nodo actúa como repetidor permitiendo encadenar nodos, dando lugar a una topología libre. La distancia máxima entre nodos es de 4.5 m. pudiendo haber hasta un máximo de 16 saltos en una cadena, alcanzando una distancia máxima de 72 m.

La alimentación de los dispositivos está incluida en la interfaz de bus, de tal forma que al conector llegan tanto los cables de datos como los de alimentación.

Conclusiones

Gran parte del esfuerzo que se está haciendo en la implantación del estándar se está centrando en conseguir imponer IEEE1394 como estándar de comunicación del PC con el resto de aparatos, intentado sustituir a los ya existentes tales como RS232, USB (véase 2.1.2.2) o SCSI. De momento solamente Sony, Apple, IBM y Compaq soportan FireWire. El mayor obstáculo que ha encontrado es que Intel de momento se niega a soportar dicho estándar debido a las tasas que Apple pide por cada dispositivo que implemente IEEE1394. Intel ha apostado por USB, lo que ha producido una masiva expansión del competidor de IEEE1394. En cambio, en el campo de la electrónica de consumo ha surgido una iniciativa llamada HAVI (véase 3.1.4) que pretende promover el uso de este bus. Otros consorcios o empresas que han adoptado IEEE1394 como estándar son: VESA (Video Experts Standards Association), Digital VCR Conference (DVC) y la European Digital Video Broadcasters. Los análisis del mercado reflejan que en principio hay cabida tanto para IEEE1394 como para USB. Se ha producido una diversificación en los segmentos de mercado de tal forma que USB está copando los periféricos lentos, aquellos que requieren una baja tasa binaria, tales como teclados, ratones o modems, mientras que su competidor se está especializando en dispositivos rápidos tales como cámaras de vídeo o reproductores de música digitales.

2.1.2.2 USB (Universal Serial Bus)

La primera especificación de USB [USB 1998] surge en 1995. La motivación del estándar, según los creadores, es la de conseguir un bus serie de alta velocidad que permita conectar fácilmente diversos periféricos a un ordenador personal.⁷ La idea es poder disponer de una gran variedad de dispositivos de bajo coste que se puedan instalar siguiendo la filosofía Plug & Play, y utilizando una única interfaz.

La arquitectura de una red USB se basa en un componente central, el ordenador personal en principio, al cual se conectan hasta 127 periféricos que se organizan formando un bus lógico. En el nivel físico hay concentradores que facilitan el despliegue de la red, siendo normal que un periférico, aparte de la función para la que está diseñado, tenga también capacidad de concentrador. La primera especificación de USB dispone de varios modos comunicación, siendo la velocidad máxima de 1,5 Mbyte/s y de 12 Mbyte/s respectivamente. Estas tasas binarias son insuficientes para poder conectar periféricos rápidos, por lo que se ha desarrollado una nueva especificación en la cual se alcanzan velocidades de hasta 240 Mbyte/s. Una de las ventajas a tener en cuenta es que los dispositivos no requieren una fuente de alimentación externa de bajo consumo, ya que el controlador del bus se encarga de proveer la alimentación a través de la conexión.

USB ha sido elegido por Microsoft e Intel como el bus de comunicación entre el ordenador personal y los dispositivos periféricos para su iniciativa EasyPC [EasyPC].

⁷ Explícitamente se menciona en las especificaciones que uno de los objetivos es facilitar la conexión del teléfono al PC

Este proyecto promueve la creación de ordenadores personales más sencillos de utilizar y configurar.

2.1.2.3 HomePNA

HomePNA [HomePNA 1998] es el acrónimo de Home Phoneline Networking Alliance, un consorcio de empresas abierto con más de un centenar de miembros. El objetivo que persigue es desarrollar una especificación que permita utilizar la red telefónica existente dentro de una casa como canal de comunicación. De momento es un estándar *de facto*.

En un principio la tasa de transmisión se ha especificado a 1 Mbps. Aunque ya se ha desarrollado tecnología para alcanzar los 10 Mbps, y se estima que la red puede soportar hasta 100 Mbps (manteniendo siempre la compatibilidad hacia atrás). Como protocolo de acceso al medio utiliza el IEEE 802.3 CSMA/CD, el mismo que las redes Ethernet (véase 2.1.3), lo cual permite reutilizar gran parte del software ya desarrollado.

Para que el protocolo HomePNA no interfiera con los servicios de voz y datos ya existentes se utiliza la técnica de FDM (Frequency Division Multiplexing), de tal forma que la voz se transmite entre 20 Hz y 3.4 KHz, los servicios xDSL entre 25 KHz y 1.1 MHz, y HomePNA a partir de 2 MHz (véase Figura 8).

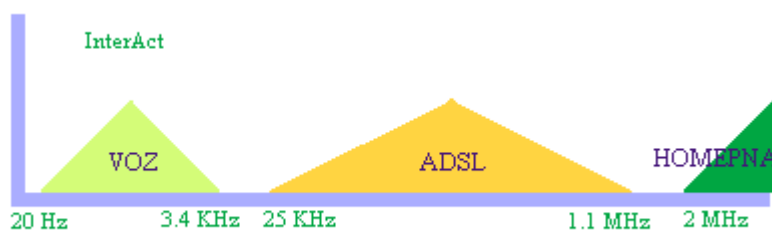


Figura 8. Clasificación de distintos tipos de tráfico multimedia según el ancho de banda requerido

En una misma red se pueden conectar hasta 25 dispositivos, por lo que se hace más apropiado como esqueleto de alta velocidad para otro tipo de redes domésticas.

Actualmente los productos disponibles incluyen PCs, impresoras, dispositivos de acceso a Internet, tarjetas de red y adaptadores USB.

Los miembros fundadores de HomePNA fueron: 3COM, AMD, AT&T, Compaq, Epigram, Hewlett Packard, IBM, Lucent, Intel, Conexant y Tut Systems.

Conclusiones

El principal problema al que se enfrenta HomePNA es que las líneas telefónicas son adecuadas para transmitir voz, pero no tráfico en tiempo real. Aun así se están realizando importantes esfuerzos para mitigar este problema.

La solución adoptada por HomePNA permite seguir utilizando el resto de los servicios que se ofrecen por la línea telefónica. Ésto, unido a la presencia de estas líneas en cualquier hogar, hace de esta tecnología un serio competidor.

2.1.2.4 Media Wire

MediaWire [MediaWire 1999] es una arquitectura de red propietaria que promueve la compañía estadounidense Avio Digital. Se fundó en 1998 con el objetivo de crear un nuevo estándar de altas prestaciones y que fuera simple de usar.

MediaWire está pensado para solucionar problemas de distribución de tráfico multimedia a lo largo de toda la casa. Utiliza cable telefónico de categoría 3 ó 5 y conectores RJ-45 como medio físico. Esto le permite tener un ancho de banda de hasta 100 Mbps que se distribuye en 32 canales de audio, ocho de MPEG-2, 16 líneas RDSI y 12 Mps restantes para control o conexiones TCP/IP.

Cada uno de ellos deberá soportar las especificaciones del protocolo. Para esa función Avio Digital comercializa dos circuitos integrados, el primero resuelve el acceso al medio, y el segundo implementa el protocolo MediaWire.

La topología se basa en una red en estrella (similar a la token ring) que asegura el ancho de banda necesario para el tráfico multimedia. El alcance entre dispositivos individuales es de hasta 33 metros con cable de CAT 3 y de 100 m con CAT5. La distancia total de la red sin encaminadores es de 4000 m. pudiéndose conectar hasta 100 dispositivos. Los dispositivos se identifican con direcciones únicas de 64 bits.

2.1.3 Red Ethernet

Es un protocolo de área local desarrollado por la empresa Xerox Corporation en cooperación con la DEC e Intel en 1976 [Metcalf, M. R. y Boggs, D. R. 1976]. Ethernet es el nombre comercial que se estableció para denominar el protocolo. Posteriormente, el estándar fue adoptado por el organismo IEEE, constituyendo las bases del protocolo IEEE 802.3. La primera publicación data de 1985, y tuvo el título de “IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and physical layer specifications.” Finalmente, el estándar fue adoptado por el Organismo Internacional de Estandarización (ISO) quedando establecido como estándar de hecho para redes de área local.

Tecnología

Dentro de una red Ethernet se pueden distinguir tres elementos característicos:

- El medio físico: según las distintas velocidades de transmisión se ha definido el estándar para tres medios físicos distintos: cable coaxial, par trenzado o fibra óptica. En la Tabla 5 se muestran las distintas combinaciones. El estándar original se planteó para una velocidad de transmisión de 10 Mbps. Para la velocidad de 100 Mbps se definió el estándar llamado Fast Ethernet, que guarda compatibilidad con el anterior. Recientemente, se ha desarrollado para 1 Gbps la denominada Gigabit Ethernet, y para marzo del año 2002 se prevé que esté completada la estandarización de la segunda parte con una velocidad de transmisión diez veces superior. Cabe resaltar que en los tres medios la comunicación se produce en banda base, de tal forma que el medio es compartido por todos los equipos del sistema.

Medio Físico	10 Mbps	100 Mbps	1 Gbps	10 Gbps
Cable coaxial	SI	NO	NO	NO
Par trenzado	SI	SI	SI	SI
Fibra óptica	SI	SI	SI	NO

Tabla 5. Relación de los distintos estándares Ethernet con los posibles medios físicos

- El protocolo de acceso al medio compartido: es un conjunto de reglas que comparten todas las estaciones de una red Ethernet. Estas reglas se

denominan CSMA/CD, y permiten gestionar el recurso compartido que es el medio. Cuando una estación quiere transmitir, primeramente comprueba que ninguna otra estación lo está haciendo. Entonces empieza a transmitir. Es posible que dos estaciones que comienzan a transmitir simultáneamente no se percaten de ello si están lo suficientemente separadas. Así, una vez que se comienza a transmitir la estación tiene que detectar si ha habido alguna colisión. En caso afirmativo intentará el reenvío después de un tiempo aleatorio.

- La trama Ethernet: Una trama consiste en un conjunto de bits organizados en varios campos. Estos campos incluyen: las direcciones de emisora y receptora, un campo de longitud variable entre 46 y 1500 bytes donde se almacena la información a transmitir, y un campo de control de errores. Si el campo de información es menor de 46 bytes, se rellena con bytes aleatorios hasta alcanzar el tamaño mínimo. Las direcciones de las estaciones Ethernet tienen una longitud de 48 bytes. Éstas son únicas en el mundo para cada estación y se preasignan cuando se fabrican. De esta forma cada estación se identifica unívocamente dentro de una red. Como las tramas son recibidas por todas las estaciones, cada una tendrá que comprobar el campo de dirección destino para saber si le corresponde la trama enviada.

Topología

En un nivel lógico una red Ethernet se puede modelar como un bus compartido por varias estaciones. En un principio el modelo de bus se correspondía con la red, de tal forma que la red estaba formada por un conjunto de estaciones que se conectaban al mismo cable coaxial. Los actuales despliegues utilizan par trenzado y se emplea un equipo especial denominado concentrador. Cada una de las estaciones se conecta a un puerto del concentrador, el cual se encarga de distribuir las tramas que le llegan por uno de los puertos a los restantes.

A cada bus diferente con sus estaciones se le denomina segmento de red. Se pueden enlazar varios segmentos para crear una red más extensa. Para ello se utilizan repetidores. Estos equipos simplemente se encargan de regenerar la señal que les llega por un segmento e introducirla en otro.

Estándares de Internet y Ethernet

Especialmente destacable es la simbiosis que existe entre estos dos mundos [Losin, P. 2000]. Por parte del IETF⁸, grupo que se dedica a regular las tecnologías utilizadas en Internet, ha existido un importante esfuerzo en utilizar el estándar Ethernet como capa de enlace. Fruto de este trabajo han sido: el desarrollo de ARP [Plummer, D. C. 1982], protocolo que se utiliza para conocer un dirección Ethernet a partir de una dirección IP, las especificaciones para cursar tráfico IPv4 [Hornig, C. 1984] y IPv6 [Crawford, M. 1998] sobre tramas Ethernet, y las especificaciones para gestionar estaciones de una red Ethernet de forma remota [Flick, J. y Johnson, J. 1999]. Esta asociación ha sido uno de los motivos para que Ethernet se haya convertido en un estándar de facto dentro del mercado de redes de área local. En la Tabla 6 se muestra una comparación de los distintos protocolos para redes LAN, en la cual se observa la elevada aceptación de Ethernet.

⁸ Internet Engineering Task Force

Tipo de LAN	Cuota de Mercado	Velocidad
Ethernet	>50%	10 Mbps
Fast Ethernet	<30%	100 Mbps
Token Ring	<18%	4/16 Mbps
FDDI	<1%	100 Mbps
Gigabit Ethernet	<1%	1 Gbps
ATM	<1%	25 Mbps/2.4 Gbps

Tabla 6. Comparación de distintas tecnologías aplicables a una red LAN según la cuota de mercado y la velocidad de transmisión. Fuente: [Orfali,R. et al. 1999]

Conclusiones

Aunque se ha comprobado el éxito notable de Ethernet en el sector empresarial, sin embargo quedaría por dilucidar cuál puede ser su futuro como red de tráfico multimedia en un hogar. El estándar parte con una experiencia de más de veinte años de desarrollo que le han consolidado como una tecnología de alta fiabilidad y elevado rendimiento. Además se pueden destacar tres importantes ventajas:

- Está soportado por un estándar internacional y ampliamente reconocido como es IEEE 802.3.
- Al estar ampliamente soportada la interoperabilidad de Ethernet con IP queda fácilmente resuelta la pasarela entre la red de interna de la vivienda y la conexión a Internet.
- Permite velocidades de transmisión desde 10 a 100 Mbps, lo cual la hace especialmente atractiva para transportar tráfico multimedia, pudiendo servir de espina dorsal (*backbone*) para otras redes más lentas.

En la parte negativa se encuentra, por un lado, que el precio de la instalación es elevado. Esto es debido al cableado que utiliza, que requiere cables de alta calidad. Además, como los hogares no poseen la infraestructura básica, hay que desplegar la red desde cero. Por otro lado, la necesidad de concentradores aumenta el precio de la instalación.

2.1.4 Redes Inalámbricas

Históricamente las redes inalámbricas han ocupado un lugar marginal dentro de la comunicación de datos. La tecnología inalámbrica era menos fiable y más cara de desplegar que el cable, quedando limitada a aquellos sectores donde el cable no podía llegar, tales como redes de satélites. Sí ha tenido en cambio un gran impacto en aplicaciones de radiodifusión, tales como la televisión o la radio, en las cuales se produce una comunicación unidireccional sin la posibilidad tener un canal de retorno. Durante la década de los años 90 hemos sido testigos de una expansión a nivel mundial de la tecnología inalámbrica aplicada a la comunicación de datos. Gracias al desarrollo de la telefonía móvil tanto analógica como digital comienza un despliegue de redes inalámbricas a nivel de cada país, creándose redes continentales. A finales de la última década, y coincidiendo con el comienzo de siglo, empezó a producirse una penetración de la tecnología inalámbrica en las redes de menor tamaño, tales como red SOHO, LAN

o redes domóticas: redes que habían sido monopolizadas anteriormente por el cable. De los varios factores claves que han influido en esta incursión destacamos dos:

- A escala mundial⁹, el éxito de la telefonía móvil ha calado en todos los estratos de la sociedad, hasta tal punto que actualmente tanto en países desarrollados como en países en vías de desarrollo, donde el despliegue de la red inalámbrica es más barato que la red cableada, el número de terminales móviles ha superado al de fijos. Ante tal expansión surge entonces la necesidad de poder realizar otros tipos de configuraciones de red más allá del modelo de telefonía móvil.
- El hecho de que en Estados Unidos es cada vez más común tener más de un ordenador en un hogar, junto con una amplia gama de dispositivos adicionales, tales como scanners, impresoras, vídeo-cameras, etc. El diseño de las casas no está preparado para tener conexiones de cable en cada estancia, ni tampoco para poder ampliar la red fácilmente, lo cual inclina hacia un despliegue de dispositivos inalámbricos. Según la consultora Strategy Analytics [Dhir, A. 2001], las previsiones de mercado arrojan que un 20 % de los hogares norteamericanos, y un 15 % de los hogares europeos tendrán una red inalámbrica como infraestructura dominante para el año 2005.

De forma más genérica la tecnología inalámbrica aporta una serie de ventajas frente al cable que se pueden resumir en dos:

- Movilidad: cuesta imaginarse nodos móviles que estén de algún modo cableados. El conjunto de aplicaciones que requieran terminales móviles se encuentra vetado al cable. La posibilidad de poder conectar cualquier dispositivo independientemente de su situación geográfica es una de las claves para conseguir una verdadera computación ubicua.
- Flexibilidad: Al contrario que las infraestructuras cableadas, que son muy costosas de reconfigurar, las redes inalámbricas son intrínsecamente muy sencillas de crear, modificar y retirar.

Dos son también los tipos de tecnologías inalámbrica que se emplean:

- Infrarroja: tradicionalmente se ha realizado un enorme despliegue de esta tecnología en aparatos de control remoto. Actualmente se están desarrollando también aplicaciones de comunicación de datos entre los periféricos y el ordenador, e incluso puede servir para el establecimiento de redes de ordenadores. Se caracteriza por utilizar la banda de infrarrojos en el espectro radioeléctrico y necesitar que los terminales tengan una línea de visión sin obstáculos para poder comunicarse.
- Radiofrecuencia: En este caso se utilizan bandas de espectro superiores en frecuencia; habitualmente se emplea la banda de 2.4 GHz, ya que esta banda está sin asignar a escala mundial. Por otro lado las técnicas de modulación también son distintas. Mientras que en infrarrojos se utiliza modulación por pulsos aquí destacan dos: espectro ensanchado y salto en frecuencia. Las comunicaciones con radiofrecuencia se pueden clasificar dependiendo del

⁹ especialmente en Europa, gracias al despliegue de la tecnología GSM

ancho de banda utilizado, en banda estrecha y banda ancha. La de banda estrecha necesita, al igual que los infrarrojos, una línea de visión clara, mientras que la de banda ancha es más flexible al poder comunicarse los terminales sin necesidad de verse.

No todo son ventajas en las redes inalámbricas, sino que también éstas adolecen de serios problemas que limitan las aplicaciones:

- **Interferencias:** el medio físico que se utiliza es el aire, donde la señal está mucho más sujeta a degradaciones que si estuviera confinada en un cable. En la tecnología infrarroja la fuente principal de interferencias es la luz solar, que contiene componentes infrarrojas que tapan la señal original. En radiofrecuencia la banda de 2.4 GHz tiene en los hornos microondas su principal enemigo. Estos dispositivos emiten pulsos aleatorios en un espectro bastante amplio de frecuencias que cubre la banda de 2.4 GHz. Además, por tratarse de una banda de frecuencias no regularizada también es sensible a que puedan surgir nuevos aparatos interferentes.
- **Disponibilidad:** un elemento básico para conseguir una calidad de servicio aceptable es que el usuario final pueda disponer de conexión de forma transparente y siempre que quiera. En este punto las infraestructuras que no utilizan cable tienen serios inconvenientes debido de nuevo al medio físico empleado. En las comunicaciones infrarrojas este factor es, si cabe, más crítico, ya que un obstáculo en la línea de visión puede cortar de forma indefinida el canal entre dos terminales. En los enlaces de radiofrecuencia, la atenuación variable del aire, tanto temporalmente como en frecuencia, puede provocar desvanecimientos aleatorios en la señal. Estas pérdidas de conexión degradan considerablemente el rendimiento del sistema, pudiendo llegar a degradar la calidad de servicio a límites inadmisibles.
- **Seguridad:** Una de las ventajas que tiene el aire frente al cable es que es mucho más sencillo compartir el medio entre los terminales. Esto hace que resulte más fácil a los intrusos intervenir el canal. Esta desventaja se acentúa en las redes de radiofrecuencia, ya que el terminal entrometido no tiene por qué estar visible al resto. Por tanto, es un requisito a tener en cuenta el proveer de mecanismos de encriptación y seguridad a los protocolos que se desarrollen para estas tecnologías.
- **Baterías:** Si el usuario quiere dispositivos realmente móviles, precisará que tengan una elevada autonomía. Cuanto más dure una batería, más espacio ocupará, dando lugar a terminales voluminosos.
- **Costo:** El desarrollo de nodos y de terminales para una infraestructura inalámbrica es menos asequible que para una cableada. Por lo expuesto hasta ahora, todos los terminales y nodos de una red han de ser lo suficientemente complejos como para poder paliar los problemas de interferencias y disponibilidad, además de tener que incorporar mecanismos de seguridad y ser de bajo consumo.

En la Tabla 7 se resumen las principales tecnologías que están disponibles en el mercado. Se clasifican siguiendo dos criterios: en primer lugar, según si el estándar es abierto, ya sea impulsado por organismos internacionales o por un consorcio de empresas, o si es propietario. En segundo lugar, atendiendo al tipo de aplicación, se tienen dos secciones diferenciadas:

- Aquellos protocolos orientados a WLAN (Wireless LAN), es decir, a redes de área local sin hilos, que serían IEEE 802.11, HiperLAN y DECT. Son redes caras pero robustas.
- Protocolos orientados al consumo residencial, mucho más ligeros que los anteriores, con peores prestaciones pero más baratos, Home RF, por ejemplo.

A mitad de camino se encuentra la iniciativa BlueTooth que surge directamente de la telefonía móvil, y se explica posteriormente.

De los protocolos de área local se ha tomado IEEE 802.11 como el más representativo, mientras que de los orientados a redes para el hogar se explica HomeRF. Por sus características especiales también se indaga dentro de BlueTooth.

Tecnología	Tipo de estándar	Max. Tasa binaria	Aplicación
Home RF SWAP 1.0	Consortio Industria	2 Mbps	Hogar, voz y datos
IEEE 802.11	Abierto (IEEE)	11 Mbps	Empresa, banda ancha, voz, vídeo y datos
BlueTooth	Consortio Industria	1 Mbps	Redes ad hoc, voz y datos
Broadband Radio Access Networks, BRAN	Abierto (ETSI)	24 Mbps	Empresa, Redes multimedia de alta velocidad
Digital Enhanced Cordless Telecommunications DECT		1.152 Mbps	Voz y datos para oficinas.
Proxim's Symphony	Propietario	1.6 Mbps	Hogar, datos.
ShareWave's Digital	Propietario	4 Mbps	Hogar, vídeo y datos
Alation's HomeCast Open Protocol	Propietario	1 Mbps	Hogar, datos
WebGear's Aviator	Propietario	1 Mbps	Hogar, datos

Tabla 7. Clasificación de distintas redes inalámbricas

2.1.4.1 IEEE 802.11

Este estándar es la propuesta [IEEE 802.11 1997] del organismo IEEE para cubrir las necesidades de una red de área local inalámbrica. Como cualquier otro estándar del IEEE, es una especificación abierta. El estándar trata de abarcar un conjunto amplio de escenarios, independientemente de que las estaciones que formen la red sean fijas, portátiles o móviles.

Una de las motivaciones principales fue abarcar el sector de telefonía celular, lo cual exigía una tasa binaria suficiente para transmitir voz empaquetada. En una primera especificación (1997) se definió una velocidad de transmisión de 2 Mbps. Posteriormente, el estándar se adecuó a las nuevas necesidades del mercado celular, en el cual empezó a surgir la idea de transmitir también vídeo. En la última especificación de 1999 ya se alcanzan velocidades de 11 Mbps.

Arquitectura

IEEE 802.11 propone dos formas distintas de configurar la red: ad-hoc o centralizada. En la primera las estaciones pueden formar una red entre ellas “al vuelo”. En la segunda, existe una estación fija que actúa de controladora, y que el resto de las estaciones toma como referencia a la hora de conectarse. Mientras que la primera opción es bastante flexible, carece de la fiabilidad que proporciona la segunda.

Independientemente de la configuración elegida el estándar define dos capas: nivel físico y acceso al medio. El nivel físico se va encargar de gestionar la transmisión de los datos entre las estaciones. Dentro de este nivel se distinguen tres técnicas distintas de establecer la comunicación: modulación por infrarrojos, espectro ensanchado y salto en frecuencia. Las dos últimas utilizan la banda de 2.4 GHz.

La capa de acceso al medio (MAC) especifica el conjunto de reglas a seguir por las estaciones, de tal forma que permite compartir el mismo medio de transmisión por todas. En esta capa se define el protocolo CSMA/CA¹⁰. Éste es una variante del protocolo utilizado en las redes 802.3. La diferencia radica en que en las redes inalámbricas no se puede detectar si ha habido una colisión a la hora de transmitir dos estaciones a la vez. Esto es debido a que cuando una estación está transmitiendo no puede escuchar si otra estación también lo está, ya que la propia señal ahoga a cualquier otra que este llegando en ese momento.

Por último, hay que destacar que IEEE 802.11 incorpora un mecanismo de seguridad que permite encriptar el tráfico que circula por la red. Para ello se define el algoritmo WEP (Wire Equivalent Privacy).

Conclusiones

IEEE 802.11 es una de las soluciones inalámbricas más completas y potentes que existe en el mercado. Como desventaja, adolece de no ser un protocolo adecuado para el entorno residencial. Debido a que IEEE 802.11 se orientó a telefonía celular, la sobrecarga que se introduce en las cabeceras es excesiva para los requerimientos de otras aplicaciones más sencillas, en las cuales el protocolo se muestra ineficiente.

2.1.4.2 HomeRF

La misión del Grupo de Trabajo HomeRF [HomeRF 2000] es la de conseguir la interoperatividad entre en el mayor número de dispositivos diferentes que estén ubicados en cualquier punto de un hogar. Para ello establece un estándar abierto y sin licencia basado en una comunicación digital mediante radiofrecuencia. El resultado ha sido el desarrollo de SWAP (Shared Wireless Access Protocol).

Home RF está compuesto de unas cuarenta empresas de las que el núcleo duro son Compaq, Ericsson Enterprise Networks, Hewlett-Packard, IBM, Intel, Microsoft, Motorola, Philips Consumer Communications, Proxim y Symbionics.

¹⁰ Carrier Sense Multiple Access with Collision Avoidance

SWAP (Shared Wireless Access Protocol)

SWAP define una interfaz común que permite soportar tanto servicios inalámbricos de voz como de datos. Ha sido diseñado para funcionar junto con la red telefónica básica y con Internet. A la hora de diseñar las especificaciones, se ha intentado reutilizar en la medida de lo posible las técnicas y parámetros más comunes de radiofrecuencia. Opera en la banda de 2.4 GHz, utilizando la tecnología de salto en frecuencia (FHSS, Frequency Hopping Spread Spectrum)¹¹ con extensiones para teléfonos móviles (DECT) y para la Ethernet inalámbrica (IEEE 802.11). Soporta también TDMA para servicios de voz en tiempo real, y CSMA/CA para servicios de datos de alta velocidad tales como TCP/IP. Las especificaciones HomeRF 1.0 admiten hasta 2 Mbps, lo cual les excluye de momento de poder enviar vídeo. El protocolo permite hasta 127 dispositivos por red y hasta 6 conexiones de voz simultáneamente.

En la parte de programación de los terminales cabe destacar que HomeRF ha apostado por la tecnología DirectX. Ésta es un conjunto de librerías desarrolladas por Microsoft para posibilitar que las aplicaciones multimedia utilicen todas las posibilidades del hardware subyacente.

La red está pensada para funcionar de dos formas: ad hoc o gestionada. En la primera todos los dispositivos tienen el mismo control sobre el medio físico. En la segunda se permiten aplicaciones que requieran una tasa mínima, ya que existe un punto de control que gestiona el medio. Este punto de control puede ser un PC conectado a la red telefónica básica o a Internet.

Actualmente las líneas de ataque de HomeRF se basan en el desarrollo de una especificación para tráfico multimedia, y en el desarrollo de una gama de productos de bajo coste y consumo.

2.1.4.3 BlueTooth

La tecnología Bluetooth [Bluetooth 2001] es una especificación abierta para comunicaciones inalámbricas de datos y voz. Las cinco compañías fundadoras del Grupo de Interés Especial Bluetooth fueron Ericsson, IBM Corporation, Intel Corporation, Nokia y Toshiba Corporation. Actualmente se han unido al grupo: 3Com Corporation, Lucent Technologies, Microsoft Corporation y Motorola Inc.

La especificación se basa en enlaces radio de corto alcance y de bajo coste, que facilitan la creación de conexiones ad hoc entre terminales tanto móviles como estacionarios.

Se utiliza la banda 2.4 GHz. La tasa binaria es de hasta 1 Mb/s, lo cual excluye el transporte de vídeo. BlueTooth se caracteriza por utilizar, en la transmisión, paquetes de pequeña longitud y un salto en frecuencia mucho más rápido que otros protocolos. Esto reduce el efecto de interferencias con otros dispositivos y mejora la transmisión en entornos ruidosos, lo que hace que sea más robusto. La distancia nominal de enlace va desde 10 cm. a 10 m., pudiéndose alcanzar los 100 m. si se aumenta suficientemente la potencia.

Todos los dispositivos BlueTooth tienen integrados potentes mecanismos de seguridad tanto de autenticación como de encriptación de las comunicaciones.

Dentro de una red Bluetooth se distinguen dos escenarios:

¹¹ Cada paquete se transmite en una frecuencia distinta de forma que se va “saltando” en la frecuencia de transmisión

- Picored, que consta de un grupo de hasta ocho dispositivos los cuales se pueden ir adhiriendo de forma aleatoria. En la picored las conexiones pueden ser entidad a entidad o establecer un maestro y que el resto actúe como esclavos.
- Red dispersa, que se forma a partir de la anterior a base de múltiples picoredes, independientes y asíncronas.

En ambos casos las conexiones pueden ser tanto punto a punto como multipunto.

Respecto a la arquitectura software definida por las especificaciones, se pueden distinguir cuatro capas bien diferenciadas por la funcionalidad de cada una:

- Núcleo de Bluetooth: es un conjunto de protocolos que se encargan de establecer el enlace entre dos dispositivos Bluetooth.
- Protocolo de sustitución del cable: se utiliza para simular el control y la transferencia de datos del protocolo RS-232. El protocolo se denomina RFCOMM.
- Protocolo de control telefónico: define las reglas para establecer y cursar llamadas telefónicas.
- Protocolos adoptados: esta capa incluye protocolos orientados a aplicación como pueden ser, PPP, TCP/IP, HTTP, FTP y WAP.

La implementación de esta arquitectura es independiente de la plataforma y del sistema operativo elegido.

Conclusiones

Aun sin poder soportar vídeo en tiempo real, todas las expectativas del mercado están puestas en Bluetooth. Las grandes empresas de telefonía móvil han elegido a Bluetooth como estándar para comunicaciones inalámbricas de corto alcance. Sin embargo, a finales del año 2000, Bluetooth todavía no había alcanzado las expectativas previstas debido a que: por un lado el precio unitario de los circuitos integrados que implementan el protocolo todavía no tiene un coste razonable como para producirlos en masa, y por otro, los escasos terminales construidos no han cumplido completamente las recomendaciones del estándar, lo cual provoca problemas de interoperabilidad entre terminales fabricados por distintas empresas.

2.2 Conclusiones

En este capítulo se ha abordado el problema de desplegar una red de datos en un hogar. Para ello se han descrito las distintas tecnologías disponibles en el mercado. Éstas van a constituir la infraestructura básica de un entorno activo.

Dependiendo del tipo de tráfico que se quiere cursar se ha realizado una primera distinción en cuanto a la red que es necesario desplegar. Por un lado están las redes de control para tráfico con velocidades de transmisión bajas, adecuados para controlar cualquier tipo de dispositivo electrónico. Por otro lado, están las redes multimedia, apropiadas para cursar vídeo y audio en tiempo real.

Dentro de las redes de control se han mostrado redes de propósito general, como son EIB y LONWork, y redes orientadas al control domótico, como son X10, CEBus y

BACNet. Las primeras se caracterizan por tener un modelo de comunicación orientado a datos, de tal forma que para la comunicación de dos dispositivos se establecen relaciones entre las variables del emisor y del receptor. En el segundo para cada dispositivo se definen un conjunto de funciones concretas que especifican las operaciones que se pueden realizar con ese dispositivo. Las prestaciones de EIB y LONWork son muy parecidas, únicamente se establece una clara diferencia en el número de dispositivos soportados. X10 es la tecnología que ofrece peor rendimiento, siendo ampliamente superada por CEBus y BACNet.

En general, todas las redes de control ofrecen diversos medios de transmisión, siendo los más comunes el par trenzado y la línea de potencia. En este caso, se establece un compromiso en la elección, en cuanto que la primera ofrece mayor fiabilidad en las comunicaciones mientras que la segunda es más barata al reutilizar el cableado ya existente.

Todas las redes multimedia estudiadas presentan mecanismos para garantizar la calidad de servicio, de tal forma que se asegure un retardo máximo en la red. Para ello los dispositivos que quieran transmitir tienen que reservar previamente una parte del ancho de banda, siendo el protocolo el encargado de decidir si se concede o no. De las redes multimedia estudiadas, IEEE 1394 y USB están más centradas en el mundo del ordenador personal, aunque en este caso IEEE 1394 pretende ser más genérico. Ambos protocolos tienen asegurada su supervivencia ya que USB se está estableciendo como el estándar de facto para comunicar periféricos lentos, mientras que la tendencia de IEEE 1394 es hacia aparatos electrónicos de audio y vídeo. En la Tabla 8 se compara la velocidad de transmisión de los distintos protocolos utilizados para comunicar dispositivos con un ordenador personal, donde cabe observar que tanto IEEE 1394 como USB quedan muy bien posicionados.

Protocolo/bus	Máxima velocidad
Puerto serie	115 kbps
Puerto Paralelo (estándar)	115 kbps
Puerto Paralelo ECP/EPP	3 Mbps
IDE	3.3 – 16.7 Mbps
Ultra IDE	33 Mbps
SSCI-1	5 Mbps
SSCI-2	10 Mbps
SSCI-3	20 Mbps
IEEE 1394	400 Mbps
USB	12 Mbps
USB 2.0	120/240 Mps

Tabla 8. Comparación de distintos protocolos de comunicación con un PC. Fuente: [Held,G. 2000]

HomePNA basa su apuesta en aprovechar el cableado telefónico existente. No es comparable con los dos protocolos anteriores ya que trata de resolver un problema distinto. IEEE 1394 y USB se plantean para conexiones próximas, menos de 4 m entre dispositivos, pudiendo alcanzar más distancia a base de intercalar dispositivos. HomePNA está más cercano a una solución de red de área local. Pretende establecerse como la espina dorsal entre las distintas redes multimedia y de control, permitiendo distribuir la información entre todas las habitaciones de la casa. No se puede saber si HomePNA triunfará en el mercado, aunque un dato a su favor es que según las previsiones [Dutta-Roy,A 1999] la línea telefónica será la solución favorita como red de

comunicación y entretenimiento dentro del hogar, frente a otras soluciones que utilicen redes inalámbricas o línea de potencia (véase Figura 9). En este punto cabe destacar la red Ethernet, que se ha impuesto en el mercado de las redes de área local y que, dada su experiencia, puede jugar un papel importante dentro del contexto de un hogar.

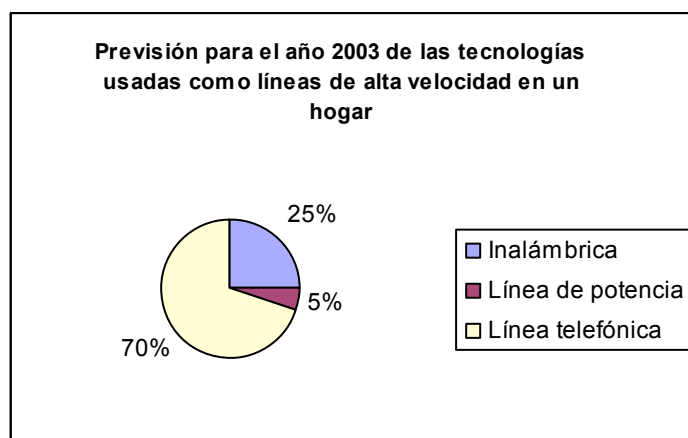


Figura 9. Previsión para el año 2003 de las tecnologías usadas como líneas de alta velocidad en un hogar. Fuente: [Dutta-Roy,A 1999]

Respecto a la tecnología inalámbrica hay que destacar dos ventajas que aporta sobre el cable: movilidad y flexibilidad. Estas características la hacen un buen complemento a las infraestructuras cableadas, y según en qué escenarios, indispensable. Se han expuesto varios tipos de soluciones, desde la más orientadas al hogar, como es HomeRF, hasta soluciones más genéricas como es IEEE 802.11. A mitad de camino se encuentra Bluetooth, una tecnología que surgió dentro del entorno de la telefonía móvil, y cuya progresión, aunque más lenta de lo previsto, la va a llevar a extenderse ampliamente.

Por último, y a modo de resumen, no existe un protocolo que cumpla con todos los requisitos tanto económicos como tecnológicos que permita definirlo como claro vencedor. Lo más probable es que en los próximos años convivirán dentro de un hogar dos tipos de redes: una de control y otra multimedia. Dentro de esta última se podrá diferenciar entre una red de alta velocidad que cumplirá funciones de distribución, y una o varias redes de conexión que podrán ser inalámbricas o cableadas.

Capa Intermedia

Un entorno inteligente se caracteriza por ser una mezcla heterogénea de numerosos componentes software y hardware. Esto obliga a solventar distintos problemas:

- Hay que manejar distintos tipos de tecnología, con el consecuente aumento de complejidad en el desarrollo. El usuario puede utilizar múltiples modos (habla, gestos, tacto, etc.) para interactuar con el entorno, con lo que se requieren interfaces de usuario muy variadas. Por otro lado la distribución de la información requiere distintos tipos de redes (véase Capítulo 2), según la naturaleza de ésta.
- La información contextual que aportan los sensores es rica en detalles pero pobre en abstracción. Las aplicaciones requieren información contextual más elaborada que la que aportan los sensores.
- Los componentes se encuentran altamente distribuidos. Tanto los sensores, que se encargan de recoger la información del entorno, como los actuadores, que transmiten la respuesta del entorno al usuario, tienen localizaciones distribuidas. Nuevamente se añade una dificultad extra a la hora de programar.
- La configuración del entorno es dinámica. No se puede prever el momento en que se conectan y desconectan nuevos dispositivos.
- El sistema tiene que estar funcionando siempre. Por ejemplo, en una vivienda un usuario consideraría inadmisibles desconexiones periódicas del

sistema. Esto implica que la gestión de nuevos componentes se tenga que realizar con el sistema ejecutándose.

- Tienen que existir mecanismos fiables de autenticación y encriptación que preserven la intimidad de usuario e impidan añadir componentes no autorizados al entorno.

Una capa intermedia es una arquitectura software que se encarga de solucionar uno o más de los problemas enunciados. Como mínimo la capa intermedia tiene que servir para presentar a las aplicaciones una visión homogeneizada de los componentes de la capa física.

Este capítulo presenta las capas intermedias de mayor impacto. Éstas se han dividido en función del ámbito de dónde surgieron, ya sea comercial o de investigación. Además las capas intermedias tienen distintas áreas de aplicación, de tal forma que se han escogido las más distintivas dentro del problema que tratan de resolver.

En el ámbito comercial se presentan: Jini, UPnP, HomeAPI, SCP, HAVi y OSGi. Las dos primeras compiten entre sí al tener enfoques similares centrados en la configuración automática de redes de área local. HomeAPI, SCP y HAVi se centran en el control de los electrodomésticos y dispositivos que se pueden encontrar en un casa. HAVi se aplica exclusivamente a los aparatos audio-visuales. Por último OSGi presenta una pasarela entre las redes exteriores e interiores a la vivienda.

En el ámbito de los proyectos de investigación se describen: Context Toolkit, Metaglué, SmartOffice, AutoHan, HomeNet. Las tres primeras se encuadran dentro del área de investigación de entornos inteligentes. AutoHan y HomeNet están más cercanas a la automatización de viviendas.

A continuación se presentan las distintas capas intermedias, siguiendo esta clasificación.

3.1 Capas intermedias comerciales

3.1.1 Jini

Jini [Edwards, W.K. 1999] es una API¹² desarrollada por Sun Microsystems. Su objetivo es convertir la red en un sistema flexible y fácil de administrar en el cual se puedan encontrar rápidamente los recursos disponibles tanto por clientes humanos como por clientes computacionales. Un entorno Jini consiste en un sistema distribuido basado en la idea de grupos federativos de usuarios y de recursos requeridos por otros usuarios. Los recursos pueden ser implementados tanto por dispositivos hardware como software.

Las partes de un sistema Jini son:

- Un conjunto de componentes que proporcionan una infraestructura de servicios federativos en un sistema distribuido.
- Un modelo de programación que soporta y estimula la producción fiable de servicios distribuidos.
- Los servicios que pueden ser parte de un sistema federativo Jini y los cuales ofrecen funcionalidad a cualquiera de los miembros de la federación.

Jini supone que la infraestructura de red que la soporta tiene suficiente ancho de banda y fiabilidad para funcionar, por lo que no aporta ningún mecanismo para mejorar estos

¹² API - Application Program Interface

dos puntos. También se asume que los dispositivos Jini tienen capacidad de procesamiento y memoria suficientes.

Existen dos conceptos claves dentro de la implementación de Jini: servicio y concesión (*leasing*).

Servicios

El concepto de servicio es el más importante dentro de la arquitectura Jini. Un servicio es una entidad que puede ser usada por una persona, un programa u otro dispositivo. Un servicio puede ser de computación, de almacenamiento, un canal de comunicación con otro usuario, un filtro software, un dispositivo hardware, o incluso un usuario. La naturaleza dinámica de Jini permite que los servicios sean añadidos o eliminados en cualquier instante de una federación, de acuerdo con las necesidades, demandas o cambios en los requisitos del grupo de trabajo que utiliza la federación. Los servicios se comunican entre si utilizando el protocolo de servicio, consistente en un conjunto de interfaces escritas en Java, que reposan sobre la tecnología de RMI.

Para conocer los servicios disponibles se utiliza el servicio de búsqueda (*lookup service*). Éste asocia las interfaces que indican la funcionalidad de un servicio con el conjunto de objetos que implementan dicho servicio. El servicio de búsqueda se organiza de forma jerárquica. Cuando se quiere añadir un servicio a la tabla se utiliza el protocolo *discovery* y el protocolo *join*. El primero se encarga de buscar la tabla de servicios (*lookup service*) y el segundo de añadir el servicio a esta tabla.

Cuando se quiere utilizar el servicio se busca en la tabla de servicios (*lookup service*) si existe. En caso de encontrarlo el cliente descarga el código de control de ese servicio almacenado en la tabla de servicios. Este código puede ser desde una simple interfaz hasta la implementación completa del servicio.

Se incorporan también un mecanismo de transacciones, para agrupar varias operaciones en una sola, así como un mecanismo de eventos.

Concesiones (*Leasing*)

El acceso a muchos de los servicios en un entorno Jini se basa en un sistema de *leasing*. Cuando se pide un servicio, éste se concede durante un periodo de tiempo determinado, tras el cual si se quiere seguir garantizando el acceso se tendrá que renovar el permiso. El tiempo de uso se negocia entre el proveedor del servicio y el cliente. La concesión puede ser exclusiva o no exclusiva.

Conclusiones

El planteamiento de Jini se centra en poder desplegar y configurar una red de dispositivos desde cero de forma automática. Los mecanismos de descubrimiento y *leasing* pretenden facilitar las tareas de añadir y eliminar dispositivos de la red. Con el concepto de servicio, se provee al programador de un mecanismo genérico y bastante flexible de poder modelar la funcionalidad de un dispositivo. Jini no ofrece una ontología de servicios, sino que deja libertad a los desarrolladores de establecer sus propias interfaces.

Jini es un producto que pertenece a la línea de desarrollo de Sun Microsystems. Ha sido confeccionado usando Java y utilizando como protocolo de comunicación RMI, lo cual supedita el éxito de Jini a la aceptación de Java y limita los desarrollos a sistemas que empleen la tecnología proporcionada por Sun. En su favor juega la enorme aprobación que está teniendo a nivel mundial el lenguaje de programación Java.

3.1.2 UPnP (Universal Plug & Play)

UPnP [Christensson,B y Larsson,O. 1999] es una arquitectura pensada para conectar entidad-a-entidad cualquier tipo de dispositivos tales como PCs, electrodomésticos inteligentes, periféricos o terminales inalámbricos. A diferencia de PnP, las comunicaciones no están centradas sobre el PC, sino que sirve para cualquier tipo de dispositivos. Es una arquitectura distribuida y abierta que se basa en los protocolos TCP/IP, UDP, HTTP y XML.

UPnP permite la configuración de los dispositivos desde cero, sin la intervención del usuario, así como el descubrimiento de nuevos recursos. El protocolo permite que se añadan dinámicamente a la red, obtengan una dirección IP y anuncien su nombre y servicios al resto. Además, un dispositivo puede abandonar la red sin dejar información indeseada.

A diferencia del modelo de Jini, que se basa en una API propietaria, UPnP se construye a partir de protocolos y estructuras de datos abiertos. UPnP es independiente tanto del lenguaje como del sistema operativo. Otra diferencia con Jini es que la arquitectura no se basa en el desplazamiento o descarga de código. La idea que subyace tras UPnP es la de una web de servicios. Ésta se consigue mediante mecanismos sencillos de descubrimiento y navegación de servicios.

Las soluciones que existen hasta la fecha para configurar una red dependen de si la red es sin servidor (entidad-a-entidad) o si está basada en servidor. En el primer caso las direcciones IP se asignan manualmente. El protocolo ARP se utiliza para relacionar la dirección IP de un dispositivo con la dirección fija y única de la red Ethernet. El descubrimiento se consigue gracias a protocolos tales como SLP (Service Location Protocol) o el SMB (Server Message Block). Para determinar si una dirección IP se puede alcanzar se utiliza el protocolo Ping o el ICMP Echo. En el segundo caso, el servidor configura la red mediante DHCP (Dynamic Host Configuration Protocol).

UPnP trabaja en ambos entornos utilizando los protocolos definidos en la Tabla 9:

	Ad Hoc	Servidor
Asignación de dirección	AutoNet (AutoIP)	DHCP
Directorio de nombre	Multicast DNS	DNS
Descubrimiento	Simple Discovery (SSDP)	Directory Service (LDAP)

Tabla 9. Protocolos empleados por UPnP

AutoNet consiste en hacer ping sobre una dirección IP arbitraria y si no responde tomarla como propia. Posteriormente, se utiliza Multicast DNS para comunicar la nueva dirección al resto de la red.

En la arquitectura de UPnP se pueden distinguir dos tipos de elementos: clientes (véase Figura 10) y objetos inteligentes (véase Figura 11).

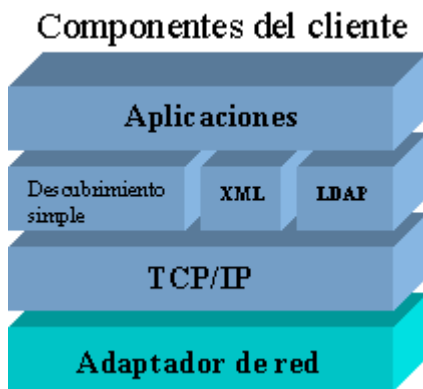


Figura 10. Componentes de un cliente UPnP

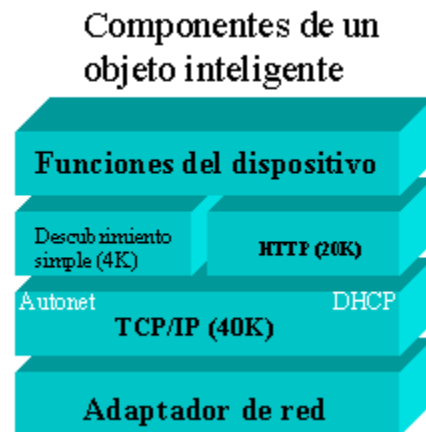


Figura 11. Componentes de un objeto inteligente UPnP

Cuando se conecta un nuevo objeto inteligente a la red se anuncia mandando un paquete multicast al resto de los dispositivos que se pueden encontrar en la red. El objeto inteligente se queda a la escucha de recibir paquetes de descubrimiento por parte de los clientes. El paquete de descubrimiento puede ser tan genérico como ¿Existe alguna impresora conectada a la red?. Si la petición del cliente concuerda con la del objeto, éste debe responder con la siguiente información: la dirección IP o el URL, la identificación del dispositivo y el identificador del paquete de descubrimiento para que el cliente sepa que su petición ha sido respondida.

Tanto los paquetes de anuncio como los de descubrimiento incluyen un enlace a un URL que contiene un fichero XML; éste se usa para describir la funcionalidad del dispositivo. Esto permite a los clientes navegar por las características del dispositivo.

El funcionamiento de UPnP se resume en la Figura 12 y la Figura 13 :

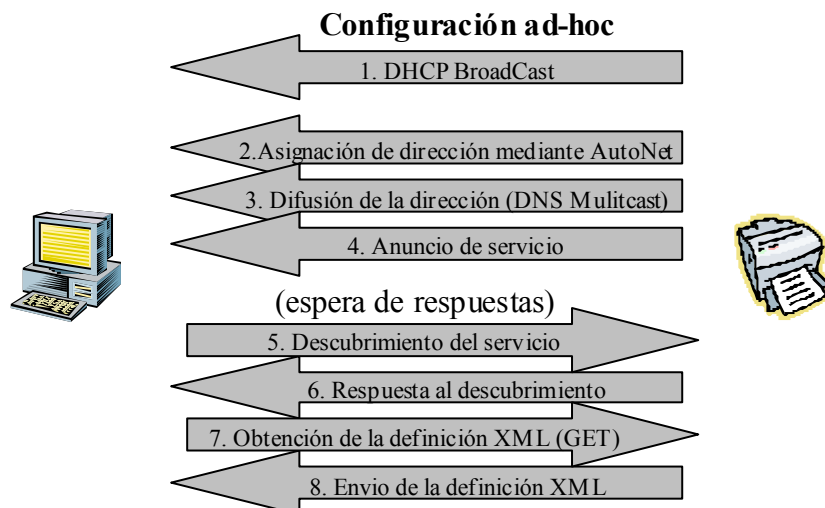


Figura 12. Configuración orientada ad-hoc de UPnP

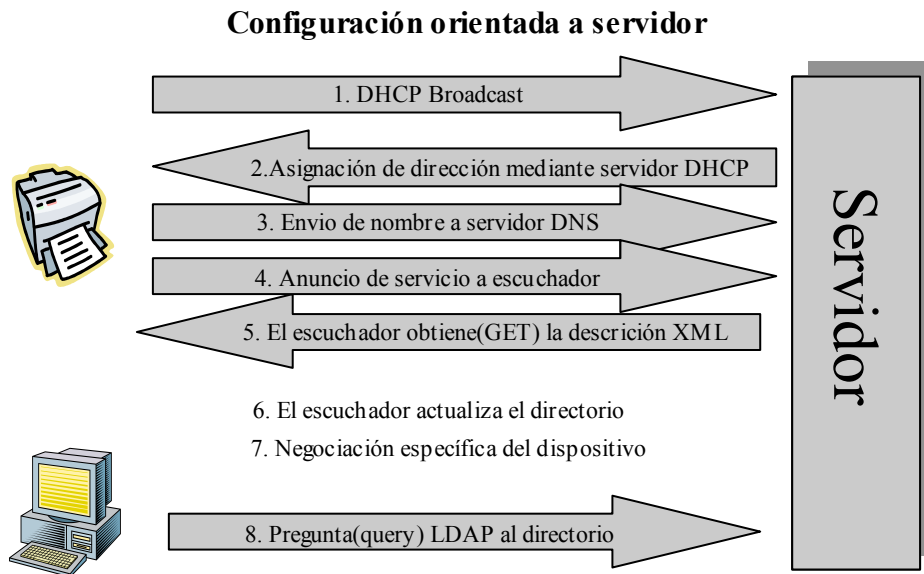


Figura 13. Configuración orientada a servidor de UPnP

Por último cabe destacar que UPnP incorpora un mecanismo de eventos denominado GENA [Cohen, J. et al 2000], basado en el protocolo HTTP. GENA extiende HTTP añadiendo tres nuevos métodos: SUBSCRIBE, UNSUBSCRIBE, y NOTIFY. Los dispositivos que quieran gestionar eventos deberán implementar un árbitro de suscripciones, que es un servidor web modificado para poder cursar también los nuevos métodos.

Conclusiones

UPnP aborda el mismo problema que Jini. En ambos se trata de automatizar la incorporación, búsqueda y eliminación de recursos de una red de área local. Jini se basa en la idea de un servidor de páginas amarillas, donde se indexan todos los servicios ofrecidos, pero junto a la entrada de cada servicio se guarda parte del código. UPnP, puede funcionar también basado en un servidor central, aunque ofrece la posibilidad de configurar la red de modo totalmente distribuido.

A nivel de funcionalidad se diferencia en que Jini incorpora mecanismos de leasing y de transacciones, que UPnP no proporciona. En cuanto a la implementación, UPnP se basa en protocolos abiertos, como son XML y HTTP, mientras que Jini se sustenta en RMI y el lenguaje Java, pertenecientes a Sun Microsystems. El éxito del lenguaje Java se ve compensado por el apoyo que Microsoft ofrece a UPnP, como sustituto de PnP. En definitiva, es bastante complicado asegurar cual de los dos va a dominar en el futuro.

3.1.3 HomeAPI

HomeAPI [HomeAPI 1999] define un conjunto de servicios e interfaces software que permiten a una aplicación descubrir y controlar los dispositivos de una red doméstica. HomeAPI pretende integrar dispositivos sencillos que hablan distintos protocolos mediante un control unificado. Quiere ser por tanto independiente de la tecnología subyacente. HomeAPI surgió en 1997, y sus miembros fundadores son Compaq Computer Corporation, HoneyWell, Intel, Microsoft, Mitsubishi Electric y Philips Electronics.

Está pensado para permitir a las aplicaciones que lo utilicen poder controlar o preguntar el estado de un dispositivo, pero no soporta el envío de flujo de datos tales como vídeo.

HomeAPI se basa en un modelo de control centralizado en el que un número pequeño de nodos inteligentes controlan un gran número de dispositivos. Cada dispositivo se modela mediante objetos OLE que tienen un conjunto de propiedades, que se pueden consultar o modificar. Un mecanismo de eventos permite avisar a las aplicaciones (nodos) suscritas de los cambios en una propiedad.

Para hacer compatible sus productos, los desarrolladores de hardware sólo tendrán que incluir junto al dispositivo el objeto OLE que lo controla, de forma que cualquier aplicación pueda acceder al mismo fácilmente a través de la interfaz. Los nuevos dispositivos también se pueden modelar a partir de objetos OLE ya definidos sin más que añadir nuevas propiedades.

Los diferentes objetos se pueden organizar de forma jerárquica utilizando contenedores de objetos que se correspondan, por ejemplo, con localizaciones dentro de la casa.

Conclusiones

La propuesta de HomeAPI no ha tenido mucha repercusión, y la iniciativa ha quedado soportada casi exclusivamente por la empresa HoneyWell. La implementación se encuentra demasiado ligada a la arquitectura OLE (actual COM), perteneciente a Microsoft, y por tanto propietaria. Por otro lado la aparición de SCP (véase 3.1.6), también soportada por Microsoft, y el desarrollo de UPnP (véase 3.1.2), han eclipsado a HomeAPI.

3.1.4 HAVi (Home Audio Visual interoperability)

HAVi [HAVi 1998] es el nombre del estándar propuesto por un consorcio formado por GRUNDIG A.G., Hitachi Ltd., Matsushita Electric Industrial Co., Ltd. Royal Philips Electronics N.V., Sharp Corporation, Sony Corporation, Thomson Multimedia S.A., Toshiba Corporation.

El estándar pretende asegurar la interconexión entre dispositivos digitales tanto de audio como de vídeo de distintos vendedores. HAVi ha adoptado IEEE1394 (véase 2.1.2.1) como capa física, por lo que se puede ver a HAVi como la capa intermedia de IEEE1394. HAVi se centra en la transferencia de audio y vídeo digital, y en las funciones de control de los dispositivos audio-visuales.

Arquitectura de HAVi

La arquitectura es independiente del sistema operativo o CPU empleada. El control de la red es distribuido, ya que cualquier dispositivo tiene autonomía para detectar, preguntar y controlar, a través de la interfaz ofrecida, a cualquier otro dispositivo, siguiendo una filosofía parecida a la de Jini (véase 3.1.1). En la práctica se distinguen dos categorías distintas de dispositivos: controladores y controlados. Para la interacción entre ambos se utilizan los denominados DCM (Device Controller Module). Éstos son elementos software que proporcionan control sobre una funcionalidad específica de un dispositivo controlado. Cada DCM ofrece una API descrita según las especificaciones de HAVi. Las DCM se ejecutan en el dispositivo controlador actuando como *proxy* del mismo. El controlador puede tener la DCM embebida o obtenerla a través de la red.

Para la interconexión de otros dispositivos no-HAVi se definen dos tipos de dispositivos: BAV (Base AV device) o LAV (Legacy AV device). Los primeros son dispositivos que implementan una parte de las especificaciones de HAVi. LAV son aquellos que ya existían antes de que se creara HAVi. Para controlar ambos dispositivos en general se necesitará un dispositivo HAVi que haga de pasarela, y que responderá a las peticiones realizadas por la DCM.

Como el control es distribuido, para poder soportar toda la funcionalidad especificada por HAVi todos los dispositivos deben tener los siguientes componentes:

- El gestor de comunicaciones con el medio físico: esta parte encapsula la dependencia de la arquitectura con la red IEEE 1394.
- El sistema de mensajes IEC 61883 [IEC 61883 1998] que especifica cómo gestionar conexiones isócronas, y un formato común de paquete de datos.
- El registro: donde se almacenan entradas para cada uno de los servicios que posee el dispositivo. Este registro puede ser consultado para realizar búsquedas de servicios.
- El Gestor de Eventos: el cual avisa de cambios en el estado del dispositivo.
- El Gestor de Recursos: que permite reservar y liberar los servicios de un dispositivo.
- El Gestor de Flujos: facilita una interfaz fácil de usar para configurar conexiones isócronas. Las conexiones pueden ser punto-a-punto o sin restricciones.
- El Gestor de DCM (Device Control Modules): que se encarga de toda la operativa que tienen asociada los DCMs.

Conclusiones

HAVi es la capa de servicios de IEEE 1394, lo que exige que cualquier desarrollo de HAVi tenga por debajo a IEEE 1394, aunque lo contrario no tiene por que ser cierto. Por tanto, el futuro de HAVi pasa por establecerse como capa intermedia de facto para dispositivos audio-visuales. Por otro lado, aunque las especificaciones suponen independencia en la plataforma elegida para implementar la arquitectura, el estándar esta íntimamente ligado al lenguaje de programación Java. Esta previsto que los dispositivos HAVi incorporen una máquina virtual Java, así como un conjunto de paquetes Java relacionados con la arquitectura.

HAVi es de todas las capas intermedias presentadas la que aporta mayor funcionalidad en la gestión de audio y vídeo, aunque lo consigue a costa de soportar un único protocolo de capa física.

3.1.5 OSGI (Open Service Gateway Initiative)

El Open Services Gateway Initiative (OSGi) [Gong, L. 2001] es un grupo de trabajo que surgió en Marzo de 1999, cuyo principal impulsor es Sun Microsystems. El resto de los miembros fundadores son: Alcatel, Cable and Wireless, Electricité de France, Enron Communications, Ericsson, IBM, Liberate Technologies, Lucent Technologies, Motorola, Nortel Networks, Oracle, Philips Electronics, Sybase y Toshiba. Actualmente tiene unos treinta socios.

El objetivo es definir y promover un estándar abierto para permitir conectar los servicios ofrecidos en redes metropolitanas (WAN) a redes de locales (LAN) o a viviendas (HAN). Esto permitirá la conexión de la próxima generación de dispositivos

inteligentes que se puedan encontrar en un hogar (oficina) con los servicios externos a la vivienda (oficina) ofrecidos a través de Internet. De esta forma, los Proveedores de Servicios de Internet (ISP), operadores de red y fabricantes de equipos pueden ofrecer una amplia gama de servicios a los usuarios finales utilizando todos la misma pasarela.

La pasarela de servicios (Services Gateway) es un servidor embebido que se implanta en la propia red para conectar la red Internet externa con los clientes internos. La SG se inserta entre la red de Proveedores de Servicios y la red interna de la vivienda, de tal forma que se separa la topología interna y externa de ambas redes. Esta SG debe ser capaz de manejar tanto flujo de datos como multimedia.

El núcleo de las especificaciones consiste en una colección de APIs que definen la pasarela de servicios. Siempre que sea posible se confiará en los estándares existentes de Java, tales como Jini o JDBC. Como la especificación OSG está orientada a la capa de aplicación, se complementa con cualquier iniciativa de las capas inferiores, entre las que se incluyen los protocolos de comunicación y redes domóticas existentes, tales como Bluetooth (véase 2.1.4.3), CEBus (véase 2.1.1.4), HAVI (véase 3.1.4), Home API (véase 3.1.3), HomePNA (véase 2.1.2.3), y HomeRF (véase 2.1.4.2).

Las especificaciones finales están disponibles desde comienzos del año 2000.

Conclusiones

OSGi es una capa intermedia cuya funcionalidad es la servir de pasarela entre dos mundos que hasta la llegada de masiva de Internet estaban separados. La aportación de OSGi radica en la estandarización de las conexiones entre los equipos de fuera y dentro de la vivienda, lo cual facilita la creación de servicios tales como voz sobre IP, TV bajo demanda, control a distancia, etc... Aunque OSGi se define como una organización independiente y sin ánimo de lucro, el hecho es que las especificaciones, al igual que HAVi (véase 3.1.4), están centradas en la tecnología Java.

3.1.6 SCP (Simple Control Protocol) – SoftWire

El Simple Control Protocol [Softwire 2000] (SCP) es un protocolo ligero a nivel de aplicación que permite descubrir y usar recursos en una red lógica. Por protocolo ligero se entiende que puede ser embebido en dispositivos hardware de bajo coste, y que utiliza un ancho de banda reducido. Una red lógica es un único grupo de dispositivos que comparte una o más redes físicas. SCP está siendo desarrollado por las empresas Microsoft y General Electric. Sin que sus aportaciones resulten demasiado novedosas, tiene la ventaja de ser un protocolo no propietario.

En el diseño de SCP se han fusionado los desarrollos del CIC (CEBus Industry Council) y de UPnP (véase 3.1.2).

Protocolo

SCP es un protocolo orientado a mensajes entre dispositivos. Cada dispositivo se modela como un conjunto de propiedades, que dan información sobre el estado, un conjunto de acciones, que aportan la funcionalidad del dispositivo, y una colección de servicios, siendo cada servicio una relación de propiedades y acciones. Los mensajes permiten consultar y modificar propiedades, así como invocar acciones del dispositivo. Se definen dos formas más de interacción: subscripciones y rutas. Las subscripciones permiten recibir notificaciones cuando cambia el valor de una propiedad. Las rutas establecen dependencias entre dos propiedades de distintos dispositivos, de tal forma que si una cambia, la otra se actualiza al mismo valor.

Todos los dispositivos tienen una dirección lógica única, independiente de la capa física. Existe un servidor central de direcciones, denominado ASA (Address Space Arbitrator). Es único para cada red, y cuando un nuevo dispositivo se conecta tiene que registrarse ante este servidor.

A nivel físico el SCP ha escogido una solución basada en la transmisión de datos por las líneas de baja tensión, SoftWire/PLC, que es similar a la que utiliza CEBus (véase 2.1.1.4). Sin embargo, también está previsto el desarrollo de varios medios físicos adicionales como el par trenzado y la radiofrecuencia.

Se ha implementado un puente a UPnP, de tal modo que cada dispositivo SCP tendrá un modelo definido por el UPnP, que consistirá en un conjunto de primitivas que permitan configurarlo.

Conclusiones

SCP, o como puede ser su nuevo nombre comercial, Softwire, es la última apuesta de Microsoft para entrar en la automatización de viviendas. Si con UPnP tiene cubierto el mercado de las redes de área local, SCP lo complementa en un sector que tiende a fundirse con el anterior. La estrategia de Microsoft es clara, abarcar ambas, y acercarlas lo más posible. De momento, SCP está en fase de desarrollo y tendrá repercusión en el mercado americano debido a las diferencias en la distribución eléctrica (véase 2.1.1.2), por lo que la llegada a Europa está todavía lejos.

3.2 Capas intermedias en proyectos de investigación

3.2.1 Context Toolkit

Ésta es una herramienta desarrollada en el Georgia Institute of Technology. Consiste en una arquitectura de objetos distribuidos para facilitar el desarrollo de aplicaciones sensibles al contexto. Según sus autores [Dey, A. K. et al 1999], se define como contexto cualquier información que pueda ser utilizada para caracterizar la situación de una entidad, donde entidad puede ser una persona, un lugar, o un objeto, tanto físico como computacional. Para completar la definición quedaría por concretar qué se entiende por situación. En este caso, la información que compone la situación de una entidad no queda claramente especificada, sino que se define como un conjunto abierto de atributos que depende de la naturaleza de la entidad. Así, si se tratara de una persona se pueden incluir la identidad, la localización en el entorno, gestos físicos que hubiera realizado, la relación de la persona con el resto de las entidades del entorno, etc. ... Si se tratara de un lugar, podrían ser relevantes la distribución espacial, el número de ocupantes, la temperatura, la presión atmosférica, etc ... En general, será el diseñador quien decida cuáles son las características más relevantes que van a definir la situación de una entidad.

Se dice que una aplicación es sensible al contexto cuando utiliza el contexto para suministrar información relevante y/o proporcionar un servicio. La motivación de la Context Toolkit es la de conseguir facilitar la adquisición y posterior utilización de información contextual. Para ello se pretende ofrecer un mecanismo genérico que presente de forma uniforme la información contextual, independientemente de la fuente que la generó. La situación de una entidad va a venir determinada por información que se recoge del mundo real, información captada por un conjunto heterogéneo de sensores. Al igual que en las redes de datos (ver 2.1), la forma de comunicarse con cada dispositivo varía enormemente. La Context Toolkit, pretende abstraer la complejidad de

tener que interactuar con distintos sensores, ya sean para medir diferentes parámetros como en el caso de sensores del mismo tipo pero comercializados por diferentes fabricantes.

Para poder abstraer los detalles de bajo nivel de la capa física, la Context Toolkit se basa en la metáfora de widget, proveniente del campo de las interfaces de usuario gráficas. Un widget es un componente software que facilita la separación de la semántica de un componente de su implementación interna. Así, un botón tiene unas características y funcionalidad definidas que para un usuario son iguales en cualquier tipo de plataforma, aunque el modo de implementarlo varíe de una a otra. Siguiendo esta línea de diseño, la herramienta presenta el concepto de widget de contexto [Salber, D. et al 1999]. Un widget de contexto encapsula en una interfaz común las características comunes de un determinado tipo de sensor. Esto permite poder reutilizar el widget aún cuando se cambie el sensor. Los widgets contextuales son intrínsecamente distribuidos, ya que las fuentes de información no se encuentran en una única máquina, sino que provienen de sensores dispersos en el entorno. Aportan ventajas similares a los widgets convencionales: facilidad de reutilización e incorporación de mecanismos de eventos. Aunque también presentan dos mecanismos que no se suelen potenciar en las interfaces de usuario:

- **Persistencia:** Los widgets de contexto no van a ser utilizados por una sola aplicación, con lo cual tienen que tener una existencia independiente. Como no se puede conocer a priori cuándo se va a solicitar la información, los widgets tienen que estar ejecutándose siempre.
- **Historia:** Independientemente de si están siendo utilizados o no, los widgets contextuales deben guardar toda la información que recolectan, ya que puede haber aplicaciones que en un futuro la requieran.

Arquitectura de la Context Toolkit

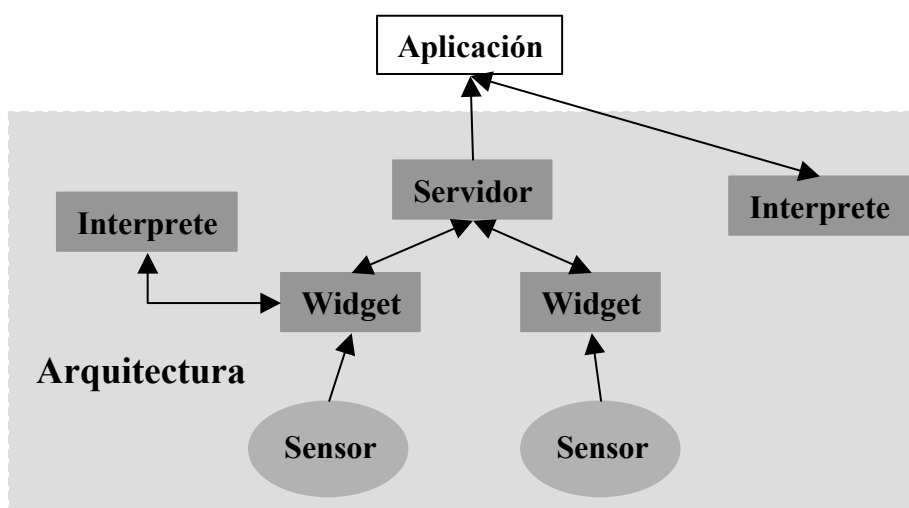


Figura 14. Arquitectura de la Context Toolkit

Tal como se muestra en la Figura 14, la Context Toolkit se compone de una arquitectura distribuida formada por tres tipos de objetos diferentes:

- **Widgets de Contexto:** un widget de contexto es un objeto que se define por un conjunto de atributos. Éstos determinan las características del widget, son tipados y pueden ser consultados mediante un mecanismo de acceso directo. Otra forma de acceder a los atributos es subscribiéndose a los cambios en el valor. El widget guarda información de la aplicaciones que se han suscrito, y de las condiciones en las que quieren ser avisadas. Así, se pueden definir umbrales sobre variables de tipo entero o real, tales como por ejemplo, “avísame cuando el atributo temperatura sea mayor que 20”. Cuando se produce una variación del valor de un atributo, no sólo se notifica, sino que también se guarda el nuevo estado del widget en una base de datos. El widget posee funciones que permiten a las aplicaciones recuperar el contexto histórico sin tener que acceder directamente a la base de datos. Por último, también se proporciona un sencillo mecanismo de llamada a procedimiento remoto.
- **Servidores de Contexto:** tal como se ha planteado el concepto de widget, éste se encuentra muy ligado a un determinado tipo de sensor. Pero la información que determina la situación de una entidad raras veces vendrá determinada por un único sensor. En este sentido los servidores se encargan de agregar distintos widgets relacionados con un entidad. Un servidor de contexto se forma a partir de todos los atributos de los widgets que agrega, de forma que cara a las aplicaciones es como si fuera un único widget. A nivel de implementación, el servidor hereda todos los métodos y propiedades de un widget, por lo que tiene las mismas funcionalidades que el anterior.
- **Intérpretes de Contexto:** representa el último nivel de abstracción. Parte de la información referente a una entidad que no se va a poder obtener directamente de la información proveniente de los sensores, sino que va a ser generada a partir de una interpretación de la información capturada por los widgets y servidores. Un intérprete de contexto consiste en un conjunto de funciones, cada una de las cuales recibe uno o varios atributos, y devuelve una nueva variable a la aplicación.

Modelo de comunicación

Volviendo a la Figura 14, en ésta no solamente se muestran los componentes de la arquitectura, sino también los flujos de información entre ellos. Se pueden distinguir tres tipos de interfaces: comunicación entre el widget y el sensor, comunicación entre los componentes de la arquitectura, y comunicación entre la aplicación y un componente. Estos dos últimos utilizan los mismos mecanismos de interconexión, que se denominarán comunicaciones de alto nivel. Las del primer tipo, en cambio, se denominarán comunicaciones de bajo nivel.

- **Comunicaciones de bajo nivel:** éstas engloban la interacción de cada widget con el sensor asociado. Como cada sensor usa protocolo distinto, estas comunicaciones serán siempre *ad hoc*, de tal forma que para cada sensor se tendrá que desarrollar un controlador específico.
- **Comunicaciones de alto nivel:** se basan en intercambio de mensajes utilizando sockets y el protocolo HTTP como capa de transporte. La implementación de la herramienta está abierta a incorporar otros tipos de protocolos basados en TCP/IP, tales como SMTP. Para la representación de los datos se emplea el lenguaje XML. Esta elección supone una integración

más ligera con la herramienta frente a otros mecanismos como pueden ser CORBA o RMI.

Conclusiones

La herramienta está siendo utilizada en un proyecto de vivienda inteligente, The Aware Home [Kidd, C. et al 1999], aunque todavía no se ha publicado una evaluación de la misma.

Resumiendo las ventajas de la Context Toolkit:

- Facilita a una aplicación el acceso a información contextual que provenga de distintas máquinas o sensores. Esto se consigue mediante los tres componentes de la arquitectura: widgets, servidores e intérpretes.
- Es multiplataforma: al haberse elegido un mecanismo de comunicación estándar (HTTP y XML), las aplicaciones que quieran utilizar la herramienta no tienen por qué ligarse a una plataforma o un lenguaje específico.
- Soporta la persistencia de los widgets de contexto, lo cual permite acceder a la información contextual siempre que se quiera.
- Guarda la evolución del contexto: posibilitando la recuperación de los cambios en los sensores, aún sin haber sido consultados en tiempo real.
- Incorpora un mecanismo de eventos, similar al de otras plataformas.

En cuanto a sus desventajas, cabe destacar que la aplicación está desarrollada en Java, de tal forma que cada widget es un objeto Java, definido por su clase correspondiente. Si se quiere realizar un verdadera distribución de los widgets se tendría que incorporar una máquina virtual en cada sensor, lo cual no puede ser soportado por la tecnología actual. Así que en la práctica la mayoría de los widgets residen en unos pocos ordenadores personales a los cuales están conectados los sensores. Como efecto secundario, las comunicaciones entre los diferentes widgets pierden parte de su ligereza al utilizar repetidamente sockets para comunicar procesos residentes en la misma máquina.

3.2.2 MetaGlue

Metaglue [Coen, M. H. et al 1999] es un lenguaje especializado para el desarrollo de sistemas interactivos y distribuidos, que ha sido desarrollado en el Laboratorio de Inteligencia Artificial de MIT. Metaglue se ha diseñado como una extensión del lenguaje de programación Java. La primera aportación de Metaglue fue la de añadir una clase Agent (Agente), que es el componente básico de cualquier sistema implementado a partir de Metaglue. Para la ejecución de los agentes se incluye una nueva máquina virtual, la Metaglue Virtual Machine (MVM). La segunda aportación es un conjunto de nuevas primitivas al lenguaje Java que van a facilitar el desarrollo de nuevos agentes.

Tanto el lenguaje Metaglue como su aproximación hacia los agentes, tienen claros referentes en proyectos realizados en este laboratorio, tales como el lenguaje SodaBot [Coen, M. 1994] y la arquitectura multiagente ScatterBrain [Coen, M. H. 1997].

La nueva máquina virtual, junto con la clase Agente y las nuevas primitivas confieren a Metaglue las siguientes características:

- Facilidades de configuración: Cada agente posee una serie de parámetros modificables, denominados atributos. Estos atributos se implementan como

un objeto *Attribute*. Metaglué posee un base de datos interna SQL en la cual se almacenan. Para acceder los atributos, se integra una interfaz web que permite modificarlos mientras se están ejecutando los agentes. Además, cuando se va a lanzar un agente se pueden configurar sus requerimientos mínimos, de tal forma que si la MVM donde se iba a ejecutar no los satisface, Metaglué busca una nueva máquina virtual que sí los cumpla.

- **Conexión de agentes:** en un primer nivel, la comunicación se resuelve mediante llamadas de un agente a los métodos de otro. En un segundo nivel, Metaglué ofrece un conjunto de capacidades abstractas que un agente puede adquirir. Por ejemplo, la capacidad de sintetizar voz, que un agente puede conseguir para luego utilizar sus métodos asociados. Las capacidades no se refieren a un agente en concreto sino que son conceptos abstractos, de tal forma que un agente no tiene que conocer quién implementa cada capacidad.
- **Mantenimiento del estado del agente:** Metaglué posee primitivas capaces de congelar el funcionamiento de un agente, pudiendo tomar de la base de datos una fotografía de los atributos en un momento dado.
- **Modificación e introducción de agentes en tiempo de ejecución:** ambas posibilidades están suficientemente cubiertas. Una de las premisas de Metaglué es mantener el sistema siempre funcionando. Si un agente termina la ejecución anormalmente, el sistema se encarga de reiniciarlo automáticamente y mantiene a la espera a los agentes que lo utilizaban.
- **Gestión de recursos compartidos:** Metaglué permite a los agentes acceder a los recursos del sistema utilizando un nivel de abstracción elevado. Los agentes no tienen que preocuparse de dónde se encuentra el recurso de ni de resolver los posibles conflictos de compartición.
- **Mecanismos de eventos:** se ha diseñado un mecanismo que permite mandar mensajes entre los agentes. Dentro de este mecanismo existe la posibilidad de suscribirse a los eventos que ofrezcan otros agentes.
- **Depuración:** Metaglué posee un interfaz gráfico que permite examinar los agentes que se están ejecutando y las interconexiones existentes.

Conclusiones

Metaglué surge porque los lenguajes tradicionales tales como C, Java o Lisp, no aportan soluciones de alto nivel para desarrollar sistemas interactivos y altamente distribuidos. Las principales aportaciones de Metaglué se pueden resumir en:

- Facilita el diseño modular y distribuido, gracias a la inclusión de la clase Agente.
- Realiza una buena gestión de recursos, que se caracteriza por permitir un acceso a los recursos con un alto nivel de abstracción.
- Permite configurar dinámicamente el sistema, además de proporcionar mecanismos de autorecuperación de los fallos que tenga.
- Incorpora herramientas de depuración para conocer el estado del sistema.

Metaglué se encuentra íntimamente ligado al proyecto de investigación denominado Hal [Coen, M. H. 1998]. Éste consiste en una habitación equipada con micrófonos, siete cámaras de vídeo, y una amplia variedad de dispositivos audio-visuales. Para controlar los equipos de la habitación se la ha dotado de aproximadamente 100 agentes Metaglué.

3.2.3 Smart Office

Smart Office [Le Gal, C. et al 1999] es un entorno inteligente desarrollado en el edificio INRIA Rhône-Alpes por el laboratorio GRAVIR-IMAG. La habitación consta de siete cámaras, que se utilizan para seguimiento de los usuarios reconocimiento de gestos y cara, ocho micrófonos para reconocimiento de voz, y una pizarra que el usuario usa para interactuar directamente con el sistema.

Arquitectura

La arquitectura de Smart Office se centra en la figura del supervisor. Éste va a servir como nexo de unión entre los distintos módulos del sistema. El supervisor se puede entender como un servidor centralizado donde se almacenan atributos y reglas sobre esos atributos, utilizando un lenguaje propietario. Los atributos y las reglas se añaden o suprimen según la decisión de los distintos módulos que interactúan con el supervisor. Mediante las reglas se pueden definir condiciones de activación de tal forma que los módulos sean avisados cuando cambie el valor de algún atributo.

Todos los módulos se comunican con el supervisor a través de sockets TCP/IP. Se distinguen dos tipos de mensajes:

- *Push*: el proceso que aporta una nueva información es el que inicia la comunicación, interrumpiendo al proceso que la recibe.
- *Pull*: el proceso que va a recibir la información toma la iniciativa de la conversación, leyendo del proceso emisor de forma no bloqueante.

En ambos casos se entiende como proceso tanto el supervisor como un módulo externo. Este tipo de comunicación se denomina de control, y puede consistir en la variación de algún atributo o en la creación/supresión de una regla. Esta información se envía a través de una red Ethernet, posibilitando la distribución de los módulos. Por otro lado, la comunicación denominada de “datos”, por un ejemplo una imagen, sólo se produce entre procesos de una misma máquina.

Conclusiones

Smart Office presenta un modelo de arquitectura centralizada, lo cual constituye un claro referente para la propuesta de InterAct (ver 4.2.1). La integración entre los diferentes módulos está orientada a recursos, de tal forma que el programador no necesita saber con qué módulos necesita interactuar, sino qué recursos requiere. Entre sus ventajas destacan las reglas, que son un mecanismo flexible que permite dotar fácilmente al sistema de nuevas funcionalidades.

En la parte negativa, cabe destacar que no existe una forma rápida y sencilla de obtener la descripción de la información del supervisor, de tal forma que el programador debe tener un conocimiento profundo sobre los recursos ofrecidos por cada módulo.

3.2.4 AutoHan

Es un amplio proyecto que se está desarrollando en el Home Area Networks Group de la Universidad de Cambridge. Este proyecto se encuentra íntimamente relacionado con los enfoques de UPnP (véase 3.1.2) y Jini (véase 3.1.1), y está claramente orientado a servir de capa intermedia de la infraestructura de red de una vivienda.

AutoHan [Saif, U. et al 2001] consiste en una arquitectura software auto-configurable cuyas características más relevantes son un registro de servicios representado en XML y un mecanismo de eventos basado en HTTP.

Las redes de control y multimedia se integran a nivel de red utilizando el protocolo IP, de forma que quedan soportados ambos tipos de tráfico. En un principio AutoHan soportaba como capa física un proyecto anterior del mismo grupo denominado Warren [Greaves, D.J. and Bradbury, ER.J. 1998]. Este proyecto trataba de utilizar redes ATM de bajo coste [Greaves, D.J. 1997] como red de un hogar. Posteriormente se han añadido HomePNA (véase 2.1.2.3), Bluetooth (véase 2.1.4.3) y FireWire (véase 2.1.2.1) como capas físicas.

Arquitectura

La arquitectura de AutoHan aporta dos funcionalidades:

- Un mecanismo de eventos basado en el protocolo HTTP. Para ello se utiliza un variante del mecanismo GENA (véase 3.1.2). Como novedad, AutoHan utiliza el protocolo UDP, en vez de TCP/IP para implementar el protocolo HTTP. Se parte de la premisa de que las redes que soportan el nivel físico son lo suficientemente fiables como para eliminar las conexiones, temporizadores y confirmaciones de TCP/IP. Además, estas últimas ya están incluidas en el protocolo HTTP, al tener cada petición asociada una respuesta.
- Un servicio de directorio de páginas amarillas denominado DHan. Permite a las entidades anunciar sus propiedades e interfaces de control, así como buscar servicios determinados. En DHan todo se describe utilizando XML: dispositivos, programas, habitaciones, personas, licencias y eventos. Cada objeto registrado tiene asignado un tiempo de concesión, que en caso de no ser renovado, causa la eliminación automática de la entrada del objeto en el directorio.

Funcionamiento de AutoHan

Todos los dispositivos generan y escuchan comandos en forma de eventos, modificando su estado en consecuencia. Para ello tienen que implementar un servidor GENA, ya sea por ellos mismo o a través de un *proxy*. Cuando se añade un nuevo dispositivo, manda un paquete multicast a la red para localizar el registro DHan. Como existe una entrada en DHan de sí mismo, éste envía la dirección IP y el puerto en respuesta al paquete UDP de descubrimiento. El nuevo dispositivo puede ahora registrar sus atributos en el DHan, entre los cuales estará la localización de su servidor GENA, y el tipo de eventos que ofrece. A partir de entonces, cualquier entidad puede buscar este dispositivo por sus atributos, o si está interesada suscribirse a alguno de los eventos ofrecidos. Para ello, DHan devuelve la dirección del servidor GENA como uno de sus atributos.

Conclusiones

AutoHan propone una arquitectura reconfigurable similar a las que proponen Jini (véase 3.1.1) y UPnP (véase 3.1.2). Incorpora elementos de ambas plataformas. Mientras que de UPnP toma la representación en XML y el mecanismo de eventos, con Jini comparte el mecanismo de “leasing”. AutoHan, tiene como valor añadido una pasarela de Internet, IHan, que facilita la integración de la arquitectura con las redes externas a la vivienda.

3.2.5 HomeNet

HomeNet [Reiter, H 1998] es un proyecto del Institute of Computer Technology de la Universidad Tecnológica de Viena. El objetivo es poder controlar los dispositivos de una casa a través de un navegador web. El resultado ha sido una herramienta que permite por un lado diseñar y configurar la red domótica independientemente del protocolo utilizado, y por otro lado acceder al control y visualización de la red a través de Internet.

La arquitectura de la aplicación resultante se muestra en la Figura 15:

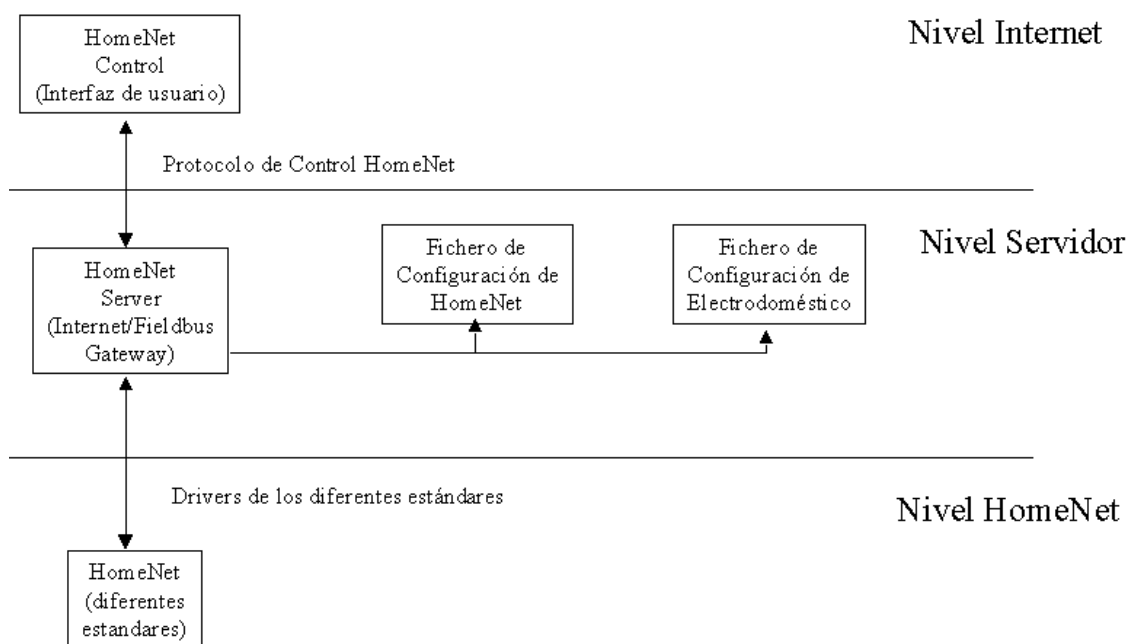


Figura 15. Arquitectura de HomeNet

La arquitectura propuesta se basa en un mecanismo de control centralizado en el servidor HomeNet. Este servidor tiene dos tipos de ficheros:

- **Fichero de configuración HomeNet:** describe mediante un lenguaje propio la estructura física de la casa (pisos, paredes, habitaciones...) y la disposición de los electrodomésticos.
- **Fichero de configuración de dispositivos:** con este fichero se abstrae el funcionamiento interno de cada dispositivo permitiendo tener varios buses de control de diferentes fabricantes.

El servidor de HomeNet se comunica con el cliente mediante un protocolo llamado HomeNet Control. La conexión se realiza a través de sockets. El protocolo permite al cliente consultar y modificar el estado de los diferentes electrodomésticos. También permite ligar electrodomésticos entre sí.

El lado del cliente se compone de una interfaz de usuario y una parte de comunicaciones (que se encarga de implementar la parte correspondiente del protocolo HomeNet Control). Gracias al fichero de configuración de HomeNet la interfaz muestra la disposición física de la casa y de los electrodomésticos de forma gráfica. De esta forma el usuario tendrá una representación visual y amigable del entorno que se ha de controlar. Con el fichero de configuración de electrodomésticos la interfaz es capaz de saber las operaciones que se pueden realizar sobre cada dispositivo. Éstas se añaden a la representación anterior para completar el lado del cliente. En resumen, gracias a la interfaz y al protocolo HomeNet, el usuario puede controlar visualmente cualquier dispositivo de la casa a través de Internet.

El mecanismo de interacción entre los dispositivos consiste en asociaciones entre variables. Cuando el valor de una variable cambia, automáticamente este cambio se transmite al de otra.

En el nivel HomeNet se encuentran los controladores (*drivers*) específicos para cada bus y que están relacionados con los niveles superiores a través del fichero de configuración de electrodomésticos. La herramienta HomeNet está programada en Java

Conclusiones

HomeNet fue un proyecto prometedor. Uno de sus últimos desarrollos fue añadir ficheros de configuración de descripciones 3D de las habitaciones. Esto posibilitaba la navegación mediante VRML. Últimamente no se han realizado avances significativos en el proyecto debido a que Heinrich Reiter, autor de la tesis doctoral que enmarcó el proyecto, abandonó la Universidad de Viena.

3.3 Conclusiones

En este capítulo se ha comprobado que el concepto de capa intermedia es bastante amplio. En general esta capa aporta dos ventajas: uniformidad y abstracción. Uniformidad porque la capa intermedia sirve de puente de unión entre las aplicaciones y los recursos o dispositivos, de tal forma que las aplicaciones acceden de la misma forma a todos los recursos disponibles, independientemente de los detalles de implementación o de la fuente de donde provengan. Abstracción, porque no solamente incluyen las funciones de nivel inferior sino que aportan al programador interfaces de más alto nivel de abstracción.

En particular cada capa intermedia contribuye con nuevas funcionalidades según donde se emplee. Se pueden establecer varias aproximaciones:

En primer lugar se tienen las denominadas arquitecturas automáticamente reconfigurables. Presentan la red como un conjunto de servicios disponibles. Para poder acceder a los servicios de la red se requieren mecanismos para describir sus interfaces de control y sus atributos, para advertir automáticamente de su presencia al resto de la red, para descubrir recursos y para interactuar con ellos. Jini, UPnP, dentro del plano comercial, y AutoHan, dentro del plano investigador, soportan las funcionalidades anteriores.

En relación con la organización de la red existen dos posibilidades: (a) orientadas a servidor, en las cuales existe un directorio, registro, o servidor de páginas amarillas, que

contiene entradas por cada uno de los servicios disponibles. (b) Ad-hoc, cada entidad de la red guarda información de los distintos servicios ofrecidos, o tiene posibilidad de descubrirlos fácilmente.

Otra característica típica de estas arquitecturas es el mecanismo de concesiones, que consiste en que un servicio se concede por tiempo determinado, pasado el cual el cliente tiene que renovar la concesión para seguir utilizándolo.

En segundo lugar, están las capas intermedias íntimamente relacionadas con la redes físicas descritas en el capítulo 2. Al contrario que las anteriores, donde el concepto de servicio es bastante genérico, estas otras se centran en el concepto de dispositivo. Dentro de este grupo se encuentran, HomeAPI, HomeNet, SPC y HAVi. HomeAPI aporta un método uniforme de crear interfaces para controlar dispositivos independiente del medio físico. HomeNet sigue la misma línea que la anterior utilizando una solución distinta, además de contemplar una representación física de la vivienda y de la situación de los dispositivos. Aunque los formatos de representación de otras capas intermedias posibilitan también modelar la geometría del entorno, HomeNet es la única que aborda el problema directamente. SPC es la solución más completa en cuanto abarca todos los niveles de la torre de protocolos OSI, desde el físico hasta el de aplicación. En los últimos niveles incluye mecanismos automáticos para añadir y descubrir nuevos dispositivos, aunque según las propias especificaciones precisa en ciertos momentos de la intervención del usuario. HAVi se restringe al mundo audio-visual. Se desmarca de los tres anteriores al aportar mecanismos capaces de manejar flujos continuos de información. Tanto HomeAPI como HomeNet están en un proceso de estancamiento, al contrario que HAVi, que se encuentra en plena expansión, y SPC, que acaba de surgir.

La tercera aproximación de capa intermedia es más heterogénea. En este caso se trata de los frutos de tres proyectos de investigación sobre entornos inteligentes, Context Toolkit, Metaglué y SmartOffice. Cada uno presenta una solución distinta, ya que trata de atacar problemas distintos. La Context Toolkit presenta una arquitectura que permite homogenizar los procedimientos de captura y distribución de la información proveniente de los sensores. Metaglué toma como marco una habitación inteligente y aporta soluciones de alto nivel para gestionar y controlar los recursos de la habitación. Smart Office, que se desarrolla en una oficina, presenta una arquitectura centrada en facilitar a los distintos módulos de aplicación un lugar común donde presentar la información que generan. Entre la Context Toolkit y Metaglué, que son los más completos, se pueden establecer coincidencias. En primer lugar ambos son distribuidos, en paralelismo al mundo real donde la información procede de distintos puntos. En segundo lugar, ambos facilitan la persistencia de los componentes del sistema, widgets en el primer caso, agentes en el segundo. Por último, ambas contribuyen con un nivel de abstracción que la capa física no es capaz de aportar, la primera en forma de interpretes de contexto, la segunda mediante el concepto de capacidades.

En un análisis más general de todas las plataformas se extraen tres apreciaciones:

Como primera conclusión, se puede observar una clara predominancia del entorno Java en la implementación de más de la mitad de las arquitecturas estudiadas, entendiéndose por entorno Java, no sólo el lenguaje sino cualquiera de los productos relacionados y distribuidos por Sun Microsystems. En la Tabla 10 se muestra la clasificación de las plataformas expuestas según su afinidad al lenguaje de programación Java.

	Comerciales	Investigación
Java	Jini, HAVi, OSGi	Context Toolkit, MetaGlue, HomeNet
No Java	UPnP, HomeAPI, SPC	SmartRoom, AutoHan

Tabla 10. Clasificación de capas intermedias según su relación con el lenguaje de programación Java

Las ventajas que aporta el lenguaje Java son conocidas: independencia de la plataforma, facilidad de aprendizaje, totalmente orientado a objetos, una amplia disponibilidad de librerías gratuitas y una gran soporte en cuanto a nuevos productos por parte de Sun. Sus inconvenientes no son menos desconocidos: la ejecución de los programas requiere gran cantidad de memoria y las aplicaciones son más lentas. Sin embargo, estas desventajas quedan parcialmente eclipsadas gracias a los rápidos avances técnicos en los equipos empleados. Esto ha ocasionado una difusión mundial del lenguaje, que queda reflejada también en las distintas iniciativas expuestas.

Por otro lado, se ha producido una respuesta de Microsoft, líder mundial en el desarrollo software, que se ha visto amenazado por el éxito de Java. En este caso Microsoft ha apostado por estándares con licencias gratuitas, ya que: (a) le sirve como hecho diferencial frente a Sun, que exige pago. (b) Las capas intermedias proporcionan la interacción entre aplicaciones, donde ya tiene asegurados cuantiosos beneficios. (c) El éxito de sistemas abiertos está ampliamente demostrado. Como resultado, Microsoft ha impulsado el desarrollo de UPnP, que se basa en estándares abiertos existentes, y va a lanzar SCP, que es un nuevo protocolo con licencia gratuita.

Como segunda conclusión, se puede observar cierta predilección por XML como lenguaje de representación. De todas las capas intermedias presentadas, tres, UPnP, Context Toolkit y AutoHan, apuestan por XML, y una, SCP, garantiza soporte de conversión. XML se utiliza para representar la información intercambiada entre las aplicaciones que utilizan la capa intermedia. Las ventajas de XML como *lingua franca* son: (a) gracias a la utilización de etiquetas la representación resultante es directamente entendible por una persona; (b) junto con las hojas de estilo, se puede mostrar en un navegador web; (c) es muy flexible ya que XML no es un lenguaje en sí, sino un conjunto de reglas para crear lenguajes; (d) es independiente de la plataforma escogida. (e) Es excelente para representar información estructurada pero irregular, gracias a que tiene un tipado débil. Como desventaja se podría destacar la ineficiencia de XML en cuanto a la relación entre los datos representados y el tamaño de las estructuras de datos utilizadas para representarlos. La alternativa que propone Java a XML es utilizar las propias clases Java como representación, y RMI, el mecanismo de llamada a procedimiento remoto de Java, para serializar los objetos y enviarlos por la red. La diferencia con la solución de Java es que XML ofrece descripciones más flexibles, ya que una aplicación no tiene por qué entender toda la estructura: aquellas etiquetas que no es capaz de procesar las desecha. Además, cabe insistir en que con XML se consigue independencia con respecto al lenguaje de programación.

Como tercera conclusión, aquellas plataformas que utilizan XML han escogido HTTP como protocolo de transporte. Esta decisión es bastante lógica teniendo en cuenta que por un lado todos los navegadores web soportan HTTP, y por otro la íntima relación de XML con HTML. HTTP es un protocolo fácil de implementar que tiene las primitivas necesarias para realizar peticiones.

Para cerrar este capítulo cabe insistir en la idea de la importancia de la capa intermedia como elemento imprescindible de un entorno inteligente. En un entorno que se caracteriza por ser altamente distribuido y heterogéneo se necesita de un elemento uniformador que facilite la interacción entre los componentes. Aun más, aunque existiera una forma de comunicarse que fuera conocida por todos los componentes,

tanto software como hardware, aun sería necesaria la capa intermedia, ya que presenta información contextual que no surge de una relación directa con los dispositivos físicos.

Arquitectura del proyecto InterAct

En el Departamento de Ingeniería Informática de la Universidad Autónoma de Madrid se ha comenzado recientemente un nuevo laboratorio dedicado a entornos inteligentes. El objetivo es desarrollar las tecnologías necesarias para llevar a la realidad tales entornos. Se ha realizado un primer prototipo del entorno inteligente: el sistema InterAct¹³. En este prototipo se plasman las ideas fundamentales sobre las que reside el diseño del proyecto.

Este primer prototipo todavía constituye la fase embrionaria del proyecto. Se piensa hacer una separación explícita entre un entorno de oficina y un entorno orientado al hogar, ya que las necesidades del usuario son sensiblemente distintas. Las interacciones que se diseñen serán específicas para cada entorno, aunque se pretende que el método de diseño y la arquitectura del sistema sean lo suficientemente flexibles como para ser reutilizados. Para ello se ha dividido el laboratorio en tres zonas: el salón, la oficina y la sala de máquinas. En esta última localización se guardan los ordenadores personales necesarios para tareas pesadas de computación, lenguaje natural y visión artificial, y el control de los diferentes dispositivos. De momento se está habilitando la infraestructura necesaria para el salón, y la sala de máquinas; dejando para una segunda fase el entorno de oficina.

¹³ Este proyecto está subvencionado por el Ministerio de Ciencia y Tecnología (TIC2000-0464).

Se ha diseñado una arquitectura en la que, siguiendo la distribución expuesta en la introducción, se distinguirían las siguientes capas: la capa física, que se ha denominado de interacción con el entorno físico, la capa intermedia o de contexto, y la capa de aplicación, donde está incluida la interfaz con el usuario.

A continuación se explican las dos primeras capas que son las que se encuentran dentro del ámbito de este documento.

4.1 La capa de interacción con el entorno físico

Dentro del laboratorio de entornos inteligentes, se tiene previsto disponer de variados dispositivos y equipos (sensores, actuadores, cámaras, micrófonos, video-proyectores, reproductores de vídeo... etc.). Tal como se expuso en el capítulo 2, existen múltiples opciones a la hora de elegir la red o redes que van a interconectar los distintos componentes. Se ha optado por realizar la misma separación que se propone en ese capítulo. Por un lado, una red de control a la que se conectan los dispositivos de automatización, y por otro lado una red multimedia para distribuir el tráfico audio-visual.

En la Figura 16 se muestra el despliegue de red realizado hasta el momento.

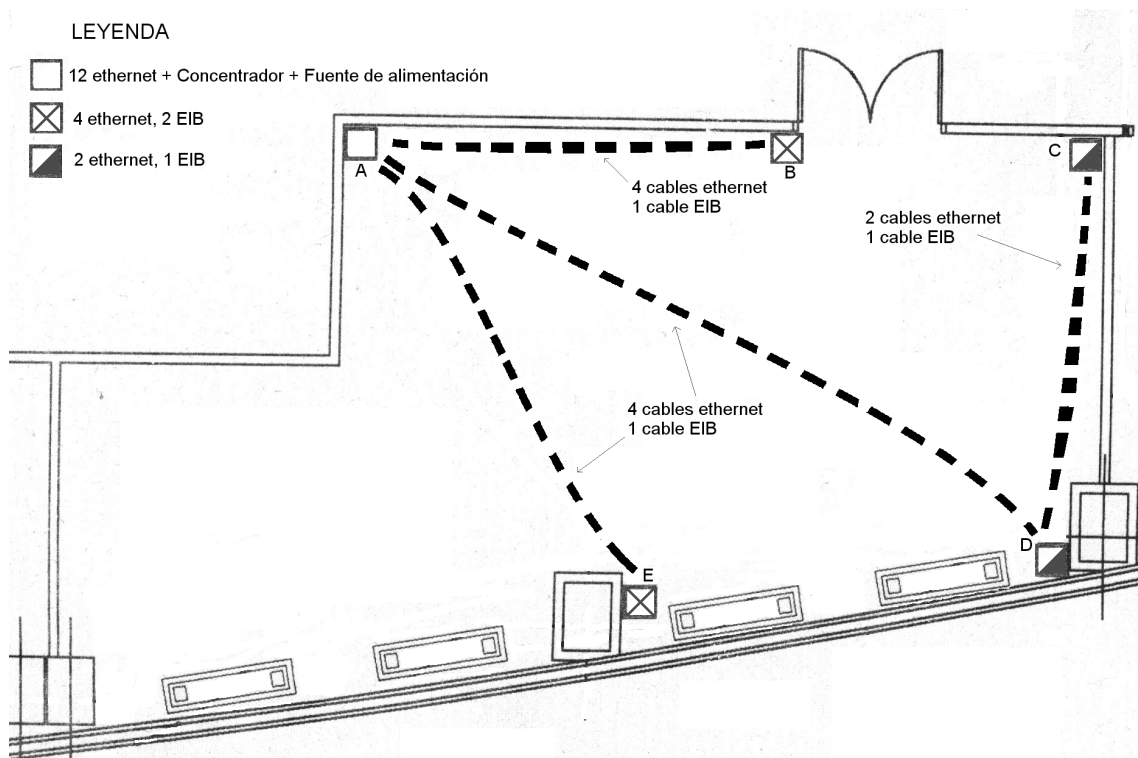


Figura 16. Plano del cableado de red del laboratorio de entornos inteligentes

4.1.1 Red de Control

Se ha elegido como red de control el bus EIB (véase 2.1.1.1). Los factores que han influido en su elección han sido:

- Es un bus que está dentro de los posibles candidatos a hacerse con el mercado europeo.
- Gracias a su difusión en Europa es fácil conseguir dispositivos.
- La oferta de dispositivos disponibles es bastante amplia.
- Es un bus bastante versátil, ya que permite poder construir dispositivos propios al tener la interfaz con el bus claramente separada.

EIB constituye el camino de conexión con los actuadores y sensores distribuidos en el entorno. Por un lado permitirá controlar los distintos dispositivos y mecanismos eléctricos que se van a conectar al entorno. Por otro lado, solucionará el problema de cómo distribuir la información capturada por los sensores. Para el primer objetivo, se ha habilitado un tipo especial de tomas de red. Estas tomas constan de un relé controlado por el bus EIB, y que actúa sobre una toma de corriente de 200 v. Esto permite actuar sobre la corriente de los electrodomésticos que se conecten, sin necesidad de modificar el dispositivo en sí. En la Figura 16 se muestra la distribución y número de tomas EIB que se han incorporado al laboratorio. Se han contemplado también actuadores para funciones concretas, como son el control del aire acondicionado, la automatización de la apertura y cierre de cortinas, y una cerradura electrónica en la puerta de entrada.

Para el segundo objetivo, se ha previsto incorporar distintos tipos de sensores que permitan capturar el estado de la habitación. En una primera fase se contempla monitorizar dos variables físicas, la temperatura y la luminosidad, y el estado de la puerta de entrada, si abierta o cerrada.

4.1.2 Red Multimedia

La segunda de las redes desplegadas será la que se encargue de transportar el tráfico multimedia. Para esta red se ha optado por la tecnología Ethernet/100 Mbps (véase 2.1.3). Varios son los factores claves en esta elección:

- Soporta tráfico multimedia: esta afirmación en principio es válida en tanto en cuanto la velocidad de transmisión es de 100 Mbps. Pero Ethernet no tiene un mecanismo que permita reservar ancho de banda, es más, al solucionar el acceso al medio mediante un algoritmo de contienda, las prestaciones de la red se degradan con el aumento del terminales. Este problema se soluciona utilizando tramos de red aislados de la red interna del Departamento, de modo que se pueda controlar el volumen del tráfico que se cursa.
- Disponibilidad: la red Ethernet se encuentra ampliamente difundida tanto en el entorno universitario como en el empresarial. Es muy fácil encontrar componentes.
- Experiencia: debido al punto anterior, se tiene bastante experiencia en el despliegue de soluciones con esta red.
- Futuro: Tal como se mostró en el apartado 2.1.3, el triunfo de la tecnología Ethernet en las redes de área local es una realidad. Este éxito probablemente se trasladará al mundo residencial.

En la Figura 16 se muestra la disposición de cableado y las diferentes tomas colocadas en el prototipo. Consta de doce tomas distribuidas en los mismos puntos que las tomas EIB. En la sala de máquinas se ha colocado un concentrador de 24 puertos que distribuye el tráfico entre todas las tomas. Está previsto utilizar un conmutador Ethernet como pasarela a la red externa del Departamento, de modo que se siga manteniendo el aislamiento con ésta.

Siguiendo con la aproximación de Coen expuesta en la introducción de este documento, se plantea utilizar cámaras de vídeo para capturar por observación directa las acciones del usuario, y una televisión de alta definición como salida de vídeo. La principal interacción del usuario con el entorno será utilizando lenguaje natural, para lo cual se dispondrá de micrófonos y altavoces. Las cámaras de vídeo se conectarán directamente a la red Ethernet, mientras que la televisión, micrófonos y altavoces, estarán ligados a ordenadores personales con adaptadores de red Ethernet.

4.2 La capa de contexto

La capa de contexto constituye la columna vertebral de la arquitectura de InterAct. Como se expuso en el capítulo tercero, un entorno inteligente está formado por conjunto heterogéneo de dispositivos físicos que recogen información del mundo real. Esta información se procesa dando como resultado un conjunto de acciones a realizar como respuesta. De acorde al conjunto de capas intermedias presentas el capítulo tercero, se pueden seguir diversos enfoques a la hora de plantearse la arquitectura sobre la cual se acomode este proceso. En nuestro caso hemos optado por una arquitectura basada en una estructura de datos centralizada en la cual se recopila toda la información proveniente del entorno físico. Este contenedor de información se denomina pizarra por tomar la idea original de los desarrollos de Hayes-Roth [Hayes-Roth, B. 1985].

Se ha denominado capa de contexto porque en la pizarra se almacena sólo información contextual, dejando la información procedimental (métodos, funciones o procesos) fuera de ésta. En la pizarra se mantiene un modelo del entorno físico y sus ocupantes; para ello se distingue entre clases e instancias. Las clases contienen la definición de los tipos de entidades integrantes del entorno, mientras que las instancias son las diferentes realizaciones que pueden tener una o más clases.

Las clases que se encuentran en la pizarra modelan el entorno físico, tanto sus integrantes como sus relaciones. Por ejemplo, habrá una instancia por cada uno de los sensores y actuadores del entorno, pero también habrá instancias y atributos que modelen aspectos de más alto nivel, como puede ser el número de personas que hay en la habitación, o la actividad que están realizando. La información en la pizarra, de alguna manera, refleja el “estado” del entorno. La organización de la información dentro de la pizarra no tiene un patrón específico, sino que se van definiendo las diferentes clases e instancias según las necesidades de cada aplicación. Uno de los objetivos a medio plazo es extraer regularidades dentro de la estructura de la pizarra, hasta conseguir una ontología que sirva como marco de referencia a la hora de desarrollar entornos inteligentes.

4.2.1 Arquitectura

La pizarra es un mecanismo de colaboración entre los agentes de aplicación: éstos son los encargados de analizar y transformar la información. Los agentes se enteran de los cambios en el mundo real leyendo directamente de las propiedades de las instancias, a cuyos cambios pueden estar suscritos. A su vez, los agentes pueden modificar las

instancias que se encuentran en la pizarra. Se definen cuatro operaciones básicas que se pueden realizar sobre la pizarra:

- GET: para obtener el valor de uno o más atributos de una instancia.
- SET: para establecer el valor de uno o más atributos de una instancia.
- SUBSCRIBE: para subscribirse a los cambios en el valor de un atributo.
- UNSUBSCRIBE: para terminar la subscripción.

En todas las interacciones con la pizarra se intercambian estructuras de datos que contienen la instancia, propiedades y valores implicados, así como el tipo de operación a realizar. Estas estructuras se definen en XML. De esta forma se independiza la implementación de la pizarra de la implementación de los módulos de aplicación, y de la de los módulos de aplicación entre sí.

La comunicación entre los módulos de aplicación y la pizarra se realiza mediante sockets, utilizando el protocolo HTTP. Esto implica que los agentes que se comuniquen con la pizarra como mínimo deberán implementar un cliente HTTP y un analizador sintáctico de XML. Además si quieren recibir subscripciones deberán añadir también un servidor HTTP.

La interfaz entre la capa física y la capa contexto se realizará también utilizando este mismo mecanismo. Actualmente se está trabajando en utilizar el protocolo SNMP para controlar todos los dispositivos conectados en ambos buses, desde e televisor a las luces, de tal forma que se incorporará un pasarela entre XML y SNMP para que todos los cambios queden reflejados en la pizarra.

Finalmente, dentro de nuestros objetivos se encuentra el poder representar tanto las clases como las instancias mediante un documento XML, de forma que la implementación de la pizarra surja de forma automática a partir de este documento XML. Para ello se está desarrollando un esquema XML y su correspondiente lenguaje, basándose en los esfuerzos del consorcio W3C [Reinhold, M. 1999] en este ámbito.

4.2.2 Flujos de información

En la pizarra se guarda la información proveniente del entorno. Este modelo no es aplicable cuando se trata de información que, o bien varía rápidamente, por ejemplo flujos de audio y vídeo, o bien tiene un tamaño excesivo, por ejemplo, archivos grandes.

El primer tipo de información no es conveniente que recabe en la pizarra, debido a las restricciones de calidad de servicio que requiere. En la pizarra lo que se modela son las fuentes y los sumideros de los flujos de vídeo y audio disponibles en el entorno, así como los canales que se establezcan entre ellos. La pizarra actúa como una matriz de conmutación a través de la cual se puede controlar el encaminamiento de todos los flujos de información continua.

En el segundo caso lo que se guarda es un enlace a la localización del archivo. Como los agentes de aplicación tienen que implementar un cliente HTTP, no les resultará complicado obtener el fichero a partir del enlace.

4.3 Primer prototipo

En una primera implementación de la pizarra se ha optado por utilizar la herramienta Context Toolkit (véase 3.2.1), ya que:

- Esta herramienta facilita la recolección de información contextual procedente de los sensores.
- El acceso a las propiedades de los widgets se realiza mediante el protocolo HTTP y XML, lo que encaja perfectamente en nuestra aproximación.
- Cubre las cuatro operaciones básicas definidas sobre la pizarra.

La pizarra internamente queda constituida por un conjunto de widgets: cada uno conserva el estado y el comportamiento de un dispositivo físico. De cara al exterior, la pizarra es accesible por los módulos de aplicación a través de un único puerto de comunicación. En un futuro próximo se obtendrá la implementación de todos los widgets de manera automática a partir del documento XML de definición de la pizarra.

El primer intento de plasmar nuestras ideas en la realidad ha dado como fruto el prototipo que se describe a continuación. El sistema implementado se compone de la capa de interacción con el entorno (véase 4.1), de la capa de contexto (véase 4.2) y de la capa de aplicación (fuera del ámbito de este documento). La capa de interacción con el entorno físico la componen interruptores, relés para controlar las luces de la habitación, un sensor de presencia y un *display*, todos ellos conectados al bus EIB. Se dispone de otro ordenador que hace de pasarela entre el bus EIB y la red de ordenadores. Posteriormente se añadirán los componentes de la red multimedia: dos cámaras, un array de micrófonos y cuatro altavoces.

La capa de contexto se compone de un widget para cada uno de los dispositivos físicos, así como de información de contexto sobre los usuarios presentes. Los widgets se distribuyen en diferentes ordenadores utilizando cada uno un puerto de comunicación para recibir y enviar mensajes. La capa de aplicación esta constituida por un gestor de diálogos en lenguaje natural que utiliza el programa ViaVoice, como reconocedor de habla y los analizadores MACO+, Relax [Carmona et al, 1998] y Tacat [Atserias, J. et al 1998]. Se ha definido un conjunto restringido de diálogos que permiten interactuar con los distintos dispositivos del bus EIB (encender/apagar las luces, preguntar si hay alguien en la habitación). [Montoro, G. 2000]

El prototipo está en funcionamiento, y va a permitir en el futuro investigar en diversas áreas de interés relacionadas con los entornos inteligentes. El objetivo es implementar el sistema final en los despachos de varios de los participantes en el proyecto, y empezar a emplearlo como herramienta de trabajo diario.

4.4 Conclusiones

En este capítulo se han resumido los desarrollos que se están realizando en entornos inteligentes dentro del Departamento de Ingeniería Informática. Se ha separado la arquitectura del sistema en tres capas: interacción con el medio físico, contexto y aplicación.

Capa de interacción con el medio físico

En la primera capa se han elegido dos tecnologías claramente diferenciadas, EIB para transmitir información de control y Ethernet para información multimedia. Se piensa distribuir sensores para capturar el estado del entorno y, siguiendo con el enfoque de Coen (véase Capítulo 1) se utilizarán cámaras y micrófonos para capturar la situación, tareas y peticiones del usuario. Es posible que parte de los sensores puedan quedar redundantes al utilizar cámaras, por ejemplo para comprobar si la puerta está abierta o cerrada, pero tres razones mantienen esta duplicidad (a) hay que dilucidar todavía el impacto social de una cámara en una habitación, (b) tener información redundante aporta fiabilidad al diseño, y (c) al tratarse de un proyecto experimental hay que mantener cualquier línea de investigación abierta.

Como trabajo inmediato en esta capa queda por desplegar toda la red y dispositivos anteriormente mencionados. En el futuro se podría plantear desplegar alguna de las redes descritas en el capítulo 2. De esta forma se comprobaría la validez de la capa contexto como uniformadora del medio físico, y se podrían realizar una comparación de los distintos protocolos. Un paso más allá en esta línea sería la implementación de una única red que aportara mecanismos de calidad de servicio, o que fuera lo suficientemente rápida para soportar tráfico multimedia, y para la cual fuera relativamente sencillo y barato diseñar dispositivos de control.

Capa de contexto

Se ha planteado una capa de contexto centralizada y basada en datos, denominada pizarra. La interfaz con la pizarra se realiza utilizando el lenguaje XML, al igual que la implementación, que surge directamente de un fichero XML donde se describen los componentes y relaciones del entorno.

Se han resaltado cuatro ventajas que aporta la arquitectura elegida:

- Facilita la cooperación entre los diversos agentes de aplicación, ya que la información compartida se encuentra centralizada.
- Debido a que XML es eficaz a la hora de describir información estructurada pero irregular, posibilita un mecanismo flexible de representación de contexto.
- Presenta a los módulos de la capa de aplicación una interfaz común que abstrae la diversidad de dispositivos del entorno físico. Cualquier tipo de red que implemente una pasarela a nuestra capa de contexto puede integrarse dentro de nuestro sistema.
- La definición y generación de la pizarra a partir de un documento XML permite facilitar el trabajo de documentación, gracias a la capacidad de formateo de texto intrínseca al lenguaje. Los desarrolladores de los módulos

de aplicación contarán con una API que incluirá toda la descripción del sistema.

- Se consigue una implementación que es independiente del lenguaje de programación elegido. Los módulos de aplicación no están atados a una plataforma específica, y la implementación del protocolo de comunicación no constituye un costo importante.
- Facilita las operaciones de depuración, ya que en la pizarra se encuentra en un momento dado toda la información sobre el estado del sistema.

La arquitectura no contempla otros aspectos tratados en las capas intermedias del capítulo 3. Por un lado no incluye mecanismo para descubrir de forma automática nuevos recursos en la red, ni presenta protocolos de autenticación y encriptación. Ambas son prestaciones que, dadas las características de la arquitectura, se deberían añadir en un futuro sin modificar la estructura interna.

Hay otras posibles líneas futuras de desarrollo. Una que está contemplada realizar es descubrir regularidades en la definición de las entidades de la pizarra, así se podrá extraer una ontología que formalice el diseño de entornos inteligentes.

Una línea que habrá que recorrer, más tarde o más temprano, es la de construir un pasarela con Internet. Gracias a la representación XML la traducción a páginas HTML será automática. También se realizará ese mismo proceso con páginas WML de forma que se pueda acceder a la pizarra desde un teléfono WAP.

Por último, otro camino interesante es el de potenciar el razonamiento y aprendizaje del sistema. Para ello, al tener una representación de las entidades y sus relaciones en la pizarra, se podrá añadir un motor de inferencia que utilice la pizarra como base de hechos.

Conclusiones y trabajo futuro

5.1 Conclusiones

Este trabajo de estudios avanzados se ha orientado hacia el área de entornos inteligentes. Se ha dividido un entorno inteligente en tres capas, centrándose el estudio en las dos primeras.

En los últimos años se ha producido un auge en el área de la automatización de viviendas. Por esta razón todas las redes de control presentadas en la capa física están orientadas en este sentido. Dentro de este campo se pueden percibir dos enfoques distintos. Por un lado, aparecen las redes orientadas al control de dispositivos que sólo se pueden encontrar en un hogar, X10 (véase 2.1.1.2), CEBus (véase 2.1.1.4) y BACNet (véase 2.1.1.7). Por el otro lado están aquellos cuya filosofía de diseño es genérica, tales como EIB (véase 2.1.1.1) y LONWork (véase 2.1.1.3). Esta distinción origina un enfoque distinto en cuanto al modelo de comunicación empleado. En los primeros cada dispositivo se describe como un conjunto de funciones que permiten controlarlo. En los segundos este enfoque ya no es válido, ya que a priori no se puede conocer la funcionalidad del dispositivo. En este caso el diseñador relaciona variables entre los dispositivos, de forma que los cambios en una variable se propagan a las variables relacionadas.

Como se vio en el capítulo 2, las redes de control son baratas y flexibles, pero tienen un ancho de banda muy limitado. Las redes que soportan tráfico multimedia son, en cambio, excesivamente caras para emplearlas como redes de control. El resultado es la coexistencia de dos tipos de redes. De las redes multimedia presentadas, IEEE 1394 (véase 2.1.2.1) y USB (véase 2.1.2.2) persiguen el mismo objetivo, aunque se ha demostrado que en la práctica cada una se orienta a un sector distinto del mercado. Ambos protocolos se encuentran en un proceso de expansión, el primero enfocado a dispositivos de alta velocidad de transmisión, tales como cámaras de video digitales y reproductores de audio y el segundo enfocado a conectar periféricos lentos tales como ratones, teclados o scanners.

Dentro de las redes multimedia hay que tener presente la solución que aporta la tecnología Ethernet (véase 2.1.3). En primer lugar debido a la simbiosis que existe entre Ethernet e Internet, lo cual posibilita desarrollar soluciones que incluyan las dos de manera muy sencilla. En segundo lugar, Ethernet ha tenido una importante repercusión en el sector de las redes de área local, y puede que su éxito se transfiera hacia entornos residenciales. Ya existe una primera tentativa en esa línea, HomePNA (véase 2.1.2.3), cuya aportación es posibilitar las comunicaciones de datos de alta velocidad utilizando la red telefónica existente en una vivienda.

Desde una perspectiva más genérica, se han observado distintas soluciones en cuanto a los medios físicos empleados. Los dos más utilizados son la línea de potencia y el par trenzado. La línea de potencia tiene la ventaja de que se puede aprovechar el cableado existente modificando solamente los dispositivos. Por contra es muy ruidosa y no existe un estándar mundial. La segunda opción obliga a tener que desplegar una nueva red, pero ofrece mejores prestaciones. Las redes de control específicas de automatización de viviendas prefieren la solución por línea de potencia, lo cual es lógico ya que es la que les permite mayor cuota de mercado.

Un medio físico que requiere un tratamiento aparte es el espacio radioeléctrico, ya que sus características son sensiblemente distintas a las del cable. Las tecnologías inalámbricas aportan dos ventajas, movilidad y flexibilidad, que las hacen indispensables para determinadas aplicaciones. Por contra conllevan una serie de problemas que implican que no siempre puedan sustituir al cable: interferencias, disponibilidad, seguridad, coste, y alimentación. De las redes inalámbricas relacionadas con los entornos inteligentes destacan tres: (1) IEEE 802.11 (véase 2.1.4.1), una solución robusta pero costosa que es adecuada para redes de área local. (2) HomeRF (véase 2.1.4.2), que es una alternativa más cercana a los planteamientos de las redes de control, aunque intenta distribuir contenidos multimedia. (3) Bluetooth (véase 2.1.4.3), que es la opción de las compañías de teléfonos móviles, bastante versátil y con un futuro prometedor.

En relación con las capas intermedias la conclusión más evidente es su necesidad. La capa física presenta una serie de problemas y carencias que hacen inevitable la inclusión de un nuevo nivel. El primer obstáculo que presenta la capa física es la variedad de redes que pueden formar un sistema: control, multimedia e inalámbrica. El desarrollador se tiene que enfrentar a una multitud de estándares y protocolos distintos, lo cual implica desarrollar soluciones específicas para cada uno y tener que disponer de pasarelas de comunicación entre todos. Este problema se acrecienta cuando se añade una nueva tecnología sobre un sistema ya implementado. En este sentido la capa intermedia sirve de elemento uniformador mediante la presentación al programador una única forma de interactuar con los dispositivos. El segundo obstáculo es la falta de abstracción. La información recopilada por los sensores está muy cercana al medio físico, lo cual hace necesario un paso intermedio que aporte mayor generalidad, y que

extraiga características que no se reflejen directamente en la información ofrecida por sensores.

De las capas intermedias presentadas, Jini (véase 3.1.1), UPnP (véase 3.1.2) y AutoHan (véase 3.2.4) aportan esta abstracción mediante el concepto de servicio. Para las aplicaciones que utilicen la capa intermedia sólo existe la noción de servicio, que es un conjunto de atributos y funciones, y es independiente de si ese servicio lo da un componente hardware o software. Estas arquitecturas están orientadas a facilitar la configuración de la red sobre la cual descansan. Para ello, presentan mecanismos que tampoco están en la capa física, como son el registro, búsqueda y eliminación de servicios.

Otras capas intermedias, como son HomeAPI (véase 3.1.3), HomeNet (véase 3.2.5) y HAVi (véase 3.1.4), se centran más en el aspecto de uniformizar el medio físico. Modelan cada tipo de dispositivo existente a partir de un dispositivo genérico sobre el cual definen las propiedades y las funciones que tiene. La diferencia entre ambas es que las dos primeras están ligadas a las redes de control, mientras que HAVi (véase 3.1.4) se sustenta sobre una tecnología multimedia, IEEE 1394 (véase 2.1.2.1), y presentando también un mecanismo de autoconfiguración.

De los cinco proyectos de investigación estudiados, dos de ellos inciden en la capa intermedia como medio de abstracción. La Context Toolkit (véase 3.2.1) presenta esta abstracción en forma de contexto, refiriéndose a la información que determina la situación de una entidad. Para ello posee mecanismo de homogeneización de la información, y un mecanismo de interpretación. En Metaglué (véase 3.2.2) la abstracción toma el nombre de capacidades. El entorno tiene una serie de capacidades genéricas, por ejemplo puede ser capaz de convertir texto en voz. Los componentes de la arquitectura pueden asumir estas capacidades de forma sencilla.

Como conclusión genérica dentro las capas intermedias se han observado dos regularidades: la primera es la pugna que mantiene Java con el resto de las tecnologías. Este enfrentamiento entre estándares propietarios y abiertos, también ha aparecido en las redes físicas, por ejemplo LONWork (véase 2.1.1.3) y EIB (véase 2.1.1.1). La segunda es la utilización de XML como lenguaje de representación. Realmente estas dos apreciaciones se funden en una, dado que las soluciones que no emplean XML utilizan Java. Acorde con estas posturas se encuentra que las que se decantan por XML terminan empleando HTTP como protocolo de transporte, lo cual es lógico dado la inmediata relación entre XML y HTTP; y en las otras, por depender Java, se sustituye HTTP por RMI.

Finalmente, se propone la arquitectura InterAct (véase Capítulo 4). Para el diseño de la capa física se ha tenido en cuenta la necesidad de tener que utilizar dos redes distintas. Como red de control se ha elegido el bus EIB, basándose en su buen rendimiento en relación con el resto de las redes, y a su disponibilidad en el mercado europeo. Como red multimedia se ha escogido la tecnología Ethernet, ya que ofrece unas elevadas prestaciones que están validadas por más de dos décadas de experiencia.

Como capa intermedia se ha diseñado una estructura centralizada, denominada pizarra, donde se recoge toda la información contextual del entorno. El resto de los componentes de InterAct utilizan la pizarra como mecanismo de cooperación. La comunicación de la pizarra con las otras dos capas se realiza utilizando estándares abiertos como son XML y HTTP. Esto hace independiente el desarrollo de la pizarra del resto de la arquitectura. La segunda aportación de XML es proporcionar una forma flexible y a la vez potente de representación. Esto permite que la implementación de la pizarra se genere a partir de un documento en XML, y viceversa.

5.2 Trabajo futuro

En un área tan incipiente como los entornos inteligentes queda aún mucho camino por recorrer. Para organizar la presentación de las posibles líneas de trabajo se ha seguido la misma división que se ha planteado a lo largo de este documento:

Capa física

Como se ha visto en el capítulo 2, la tecnología actual parece implicar la coexistencia de dos redes: una para el control y otra para la información multimedia. Una dirección de investigación interesante es la de conseguir una única tecnología que permita realizar la fusión entre ambas redes. Esto supondría una reducción de los costes y una mayor simplicidad de diseño, al tener que desplegar un único tipo de red. Se plantean dos aproximaciones al problema:

- Mejorar las prestaciones de las redes de control para que incorporen mecanismos que permitan garantizar el ancho de banda. Hay que tener en cuenta que no es suficiente con aumentar la velocidad de transmisión, ya que un número suficiente elevado de dispositivos acabaría por degradar el sistema. Se necesitan además mecanismos para la reserva de ancho de banda para procesos concretos.
- Abaratar los costes de las redes multimedia. Éstas poseen los mecanismos necesarios para transmitir tráfico de control, pero si un dispositivo quiere conectarse a la red, aunque sólo sea para recibir información de control, tiene que pagar un alto coste tanto monetario como computacional, ya que los protocolos utilizados son bastante complejos. Se tendría, pues, que indagar en conseguir simplificar los protocolos utilizados sin perder prestaciones, o alternativamente que la red fuera capaz de discernir entre dispositivos que cursan distinto tipo de tráfico.

Siguiendo al hilo del punto anterior, aunque la red soportara tráfico de control, se seguiría teniendo el problema de que los dispositivos audiovisuales no están preparados para recibir dicho tráfico de control. En efecto, muchos de estos dispositivos han pertenecido tradicionalmente al campo analógico, tales como micrófonos y altavoces. Aquí hay que incidir en la creación de dispositivos que cumplan las especificaciones de estándares como IEEE 1394 y HAVi.

Dentro de las redes de control, pensando a más corto plazo, se pueden acometer distintas mejoras. En general, aunque todas presumen de soportar una amplia variedad de medios físicos y tecnologías, entre los que se incluyen fibra óptica y Ethernet, todavía tienen que aparecer soluciones reales distintas de las habituales: la línea de potencia y el par trenzado.

Por su parte a la tecnología inalámbrica tiene dos problemas a solucionar dependiendo del ámbito de aplicación: si los nodos son fijos, pero utilizan enlaces de radiofrecuencia, el principal problema son las interferencias, ya que se emite en bandas sin regularizar. Se requiere un esfuerzo de reglamentación para evitar conflictos. Si las estaciones son móviles, al problema anterior se añade el de la alimentación y el tamaño. Se están realizando grandes esfuerzos para conseguir cada vez terminales con mayor autonomía y menor tamaño.

Capa intermedia

Una característica que comparten la mayoría de las capas intermedias es que tienen la capacidad de autoconfigurar las entidades que componen la red. Éstas permiten que se añadan recursos dinámicamente, anunciando cuándo ocurren tales eventos para que otras entidades puedan utilizarlos. Este enfoque, que en principio parece apropiado, deja sin solucionar un problema crítico. Se exige que las entidades del sistema tengan un conocimiento previo de todos los posibles recursos del sistema, o que sean capaces de entender la semántica de una nueva interfaz. En el primer caso es necesario un preacuerdo por parte de los desarrolladores en el que se establezcan todas las posibles interfaces. En el segundo, se requiere un mecanismo que permita comprender sobre la marcha el significado de la nueva interfaz. En ambos casos existe una carencia de herramientas para facilitar el desarrollo. Las soluciones que se han tomado hasta ahora son de dos tipos:

- Dejar totalmente libre la descripción de las interfaces. El problema es que en este caso no se aportan facilidades para que los desarrolladores puedan conocer las interfaces disponibles sobre las cuales realizar sus desarrollos.
- Especificar todas las interfaces posibles. Como los campos de aplicación son muy amplios, el resultado son interfaces específicas y ligadas a los dispositivos físicos. Se precisarían descripciones más abstractas que incluyan conceptos desligados de la capa hardware.

En el primer caso, el sistema funciona bien cuando la interfaz está orientada al usuario, de forma que el que tiene que interpretarla es una persona. Una línea que tienen que indagar las anteriores arquitecturas es la de investigar en lenguajes y ontologías que permitan a los componentes software tener una comprensión de la interfaz que están utilizando.

Como punto final, y en relación con el camino abierto por OSGi (véase 3.1.5), todavía queda bastante camino por recorrer en la implementación de una pasarela entre las redes internas y externas a la vivienda. En este sentido se está todavía trabajando en formalizar los servicios ofrecidos de la residencia hacia el exterior.

Proyecto Interact

Dentro del proyecto InterAct se van a atacar algunos de los problemas mencionados. Entre ellos cabe destacar:

- Unificar las dos redes físicas empleadas mediante un mecanismo de abstracción.
- Diseñar e implementar la arquitectura de pizarra propuesta cubriendo los siguientes puntos:
 - Comunicación y descripción en XML.
 - Mecanismos de notificación.
 - Generación de información de alto nivel de abstracción, íntimamente relacionada con el contexto.
 - Implementación distribuida de la pizarra.
 - Incorporación de mecanismos genéricos para almacenar y consultar la historia del contexto.

- Incorporación de un mecanismo que permita controlar adecuadamente los flujos de información multimedia de la habitación.
- Proponer una ontología de la información contextual a partir de regularidades en la información que se almacene en la pizarra.

Agradecimientos

El proyecto InterAct es el fruto del trabajo de varios integrantes del Departamento de Ingeniería Informática, aunque en este documento los resultados hayan quedado recopilados por una sola persona. Por orden alfabético los integrantes actuales son: Xavier Alamán, José Alarcó, Eloy Anguiano, Francisco Gómez, Antonio Martínez, Javier Martínez y Germán Montoro.

También quiero agradecer al Grupo Future Computing Environments del Georgia Institute of Technology por haber facilitado la herramienta Context Toolkit.

Por último, agradezco a Germán sus acertadas correcciones, que añadidas a las de mi tutor han hecho este texto algo más comprensible.

Bibliografía

Atserias, J. et al 1998. Morphosyntactic Analysis and Parsing of Unrestricted Spanish Text, In *Proceedings of the 1st International Conference on Language Resources and Evaluation* (LREC'98, Granada, Spain).

Bluetooth 2001. Specification of the Bluetooth System. Core. Bluetooth Special Interest Group

CAL1 1996. EIA-600.81 Common Application Language. Electronic Industries Association

CAL2 1996. EIA-600.82 Common Application Language. Electronic Industries Association

Carmona, J. et al, 1998. An Environment for Morphosyntactic Processing of Unrestricted Spanish Text. In *Proceedings of 1st International Conference on Language Resources and Evaluation* (LREC'98, Granada, Spain).

CEBus 1995. EIA-600.10 Introduction to CEBus Standard. Electronic Industries Association

Christensson, B y Larsson, O. 1999. Universal Plug and Play Connects Smart Devices. In Windows Hardware Engineering Conference.

Coen, M. 1994. SodaBot: A Software Agent Environment and Construction System. AI Lab Technical Report 1493. Massachusetts Institute of Technology. Cambridge, MA

Coen, M. H. 1997. Building Brains for Rooms: Designing Distributed Software Agents. In *Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence (IAAI 97)*.

Coen, M. H. 1998. Design Principles for Intelligent Environments. In *Proceedings of the 1998 National Conference on Artificial Intelligence (AAAI-98)*

Coen, M. H. et al 1999. Meeting the Computational Needs of Intelligent Environments: The Metaglu System. In *Proceedings of the Managing Interactions in Smart Environments (MANSE'99)*

Cohen, J. et al 2000. General Event Notification Architecture Base: Client to Arbitrator. Work in progress, Internet draft.

Crawford, M. 1998. Transmission of IPv6 Packets over Ethernet Networks. RFC 2464, Fermilab.

Dey, A. K. et al 1999. A Context-Based Infrastructure for Smart Environments. In *Proceedings of the Managing Interactions in Smart Environments (MANSE'99)*.

Dhir, A. 2001. *The Home Networking Revolution, A Designer's Guide*. Xilinx Inc.

- Douligeris, C. 1993. Intelligent Home Systems. In IEEE Communications Magazine. pp 52-61
- Dutta-Roy, A. 1999. Networks for Homes. IEEE Spectrum. Diciembre. pp 26-33
- EasyPC <http://www.easypc.org>
- Edwards, W.K. 1999. Core Jini. The Sun Microsystem Press, Java Series.
- EHS European Home Systems Specification. European Home Systems Association.
- Fernández-Valdivieso, C. et al 2000. EIB remote control through applets and mobile phones. In Proceedings EIB Scientific Conference 2000.
- Fernández-Valdivieso, C. et al. 2000. Herramienta Software para el Control Remoto de Instalaciones Domóticas con Bus EIB. En Libro de Actas de XV Simposium Nacional de la Unión Científica Internacional de Radio, URSI 2000.
- Flick, J. y Johnson, J. 1999. Definitions of Managed Objects for the Ethernet-like Interface Types. RFC 2665, Hewlett-Packard Company y RedBack Networks.
- Gong, L. 2001. A Software Architecture for Open Service Gateways. IEEE Internet Computing. Jan-Feb.
- Goossens, M. 1998. The EIB System for Home & Building Electronics. EIBA s.c.
- Greaves, D.J. 1997. ATM in the Home – and – The Home Area Network. Presented at IEE Colloquium on ATM in Professional and Consumer Applications, Savoy Place, London.
- Greaves, D.J. and Bradbury, ER.J. 1998. Warren: A Low-Cost Home Area Network. IEEE Network, Vol 12, No. 1, Jan., pp. 44-56
- HAVi 1998. The HAVi Specification – Specification of the Home Audio/Video Interoperability Architecture.
- Hayes-Roth, B. 1985. A Blackboard Architecture for Control. *In Artificial Intelligence*. pag 251-321
- Held, G. 2000. Understanding data communications from fundamentals to networking. John Wiley & Sons
- HomeAPI 1999. Home Applications Programming Interface White Paper. HomeAPI Working Group.
- HomePNA 1998. Simple, High-Speed Ethernet Technology for the Home. The Home PhoneLine Networking Alliance.
- HomeRF 2000. Introduction to Home RF Technical Specification. HomeRF Working Group.

Hornig, C. 1984. A Standard for the Transmission of IP Datagrams over Ethernet Networks. RFC 894, Symbolics Cambridge Research Center.

IEC 61883 1998. IEC 61883, Consumer Audio/Video Equipment-Digital Interface, parts 1-5, Int'l Electrotechnical Commission, Ginebra, Suiza.

IEEE 1394 1995. High Performance Serial Bus. IEEE

IEEE 802.11 1997. Wireless LAN Medium Access Control and Physical Layer Specification. The Working Group for WLAN Standards

Kastner W., y Krügel C. 2000. Jini Connectivity for EIB Home and Buildings Networks: From Design to Implementation. In EIB-Proceedings Contributions part 3/2000

Kidd, C. et al 1999. The Aware Home: A Living laboratory for ubiquitous computing research. In The Proceedings of CoBuild'99

Le Gal, C. et al 1999. Smart Office: An Intelligent and Interactive Environment. In Proceedings of the Managing Interactions in Smart Environments (MANSE'99)

LONWork 1999. Introduction to the LONWork System. Version 1.0. Echelon Corporation.

Losin, P. 2000. Essential Ethernet Standards. RFCs and Protocols Made Practical. Wiley Computer Publishing

Maier, M. W. et al 2001. Software Architecture: Introducing IEEE Standard 1471. IEEE Computer, Vol. 34, No. 4, Apr. pp. 107-109

MediaWire 1999. MediaWire Technology WhitePaper. Avio Digital.

Metcalf, M. R. y Boggs, D. R. 1976. Ethernet: Distributed Packet Switching for Local Computer Networks. In Communications of ACM, Vol 19, No 5, pp. 395-404.

Montoro, G. 2000. Entornos Inteligentes: Un nuevo paradigma de interacción. Departamento de Ingeniería Informática, UAM.

Newman, H.M. 1997. BACnet – The New Standard Protocol. Electrical Contractor. pp. 119-122

Orfali, R. et al. 1999. Client/Server Survival Guide. Wiley Computer Publishing. 3ª Edición

Ortega, et al 2001. Nuevos paradigmas de interacción en el aula del siglo XXI. En las Actas del 2º Congreso Internacional de Interacción Persona-Ordenador (Interacción' 2001). Salamanca

Palensky, P. 2000. Intelligent Software Agents for EIB Networks. In EIB-Proceedings Contributions part 3/2000

Pentland, A. 2000. Perceptual Intelligence. In Communications of the ACM. Vol. 43, No. 3, March.

Pentland, A. 2000. Perceptual Intelligence. Communications of the ACM. Vol, 43, No, 3, March.

Plummer, D. C. 1982. An Ethernet Address Resolution Protocol, or, Converting Network Protocol Addresses to 48-bit Ethernet Address for Transmission on Ethernet Hardware. RFC 826, MIT.

Reinhold, M. 1999. An XML Data-Binding Facility for the Java Platform. Core Java Platform Group, Java Software.

Reiter, H. 1998. Internet Connectivity for Residential Field Area Networks. Tesis doctoral de la Technische Universität Wien Technisch-Naturwissenschaftliche Fakultät

Reiter, H. 1998. The Internet Challenge: Establishing Global Connectivity for EIB Networks. In EIB-Proceedings Contributions part2/1998

Saif, U. et al 2001. Internet Access to a Home Area Network. IEEE Internet Computing. Jan-Feb.

Salber, D. et al 1999. The Context Toolkit: Aiding the Development of Context-Enabled Applications. In The Proceedings of CHI'99 Software 2000. Software Protocol Specification. Microsoft Confidential, Preliminary Version 0.72

Tanenbaum, A. S. 1996. Computer Networks. Third Edition. Prentice Hall
USB 1998. Universal Serial Bus. Revision 1.1

Warriner, P. y Karam, K. Z. 1998. NUDAN – A Multifunctional Home Automation Network. In ICCE 1998 Conference.

Weiser, M. 1991. The Computer for the 21st Century. Scientific American. Pp. 94-100, Sept.

X10 The X10 Specification, X-10 (USA) Inc, 91 Ruckman Rd., Box 420, Closter, NJ 07624

Apéndice I.

Ejemplo de la pizarra utilizando la Context Toolkit

En este apéndice se muestra el código necesario para construir una implementación simple de la pizarra utilizando la Context Toolkit. El ejemplo propuesto consiste en dos widgets, uno que representa a una bombilla y otro a un interruptor. Los widgets tienen cada uno un atributo, *status*, que en el primero representa el estado de encendido/apagado, y en el segundo el estado activo/no activo. También tienen una subscripción, *update*, que avisa a los subscriptores de cuándo ha cambiado el atributo *status*.

En la Figura 17 se muestra la relación entre los componentes del sistema. Las líneas discontinuas muestran las subscripciones, las líneas continuas transferencia de datos directos.

La comunicación entre los widgets y los agentes se realiza mediante XML y HTTP, utilizando las facilidades que aporta la Context Toolkit. Para la comunicación entre los dispositivos y la pizarra se ha implementado una pasarela al bus EIB utilizando Visual Basic.

Se han implementado dos agentes. El agente A está suscrito a los cambios del interruptor. Cuando el interruptor se activa, el agente A refleja el cambio en la bombilla cambiando el estado del widget que representa a la luz. El widget B está suscrito a los cambios de ambos widgets, de forma que monitoriza todas las novedades que se producen en la pizarra.

Si se añadiera un nuevo dispositivo, por ejemplo un detector de presencia, el agente se podría subscribir a los cambios del detector y provocar que la luz también respondiera a la presencia de las personas. También se podría plantear que en vez de que el agente A encendiera la bombilla, pusiera en marcha un motor que controle las persianas. En todas estas modificaciones lo único que hay que hacer en la pizarra es añadir nuevas descripciones.

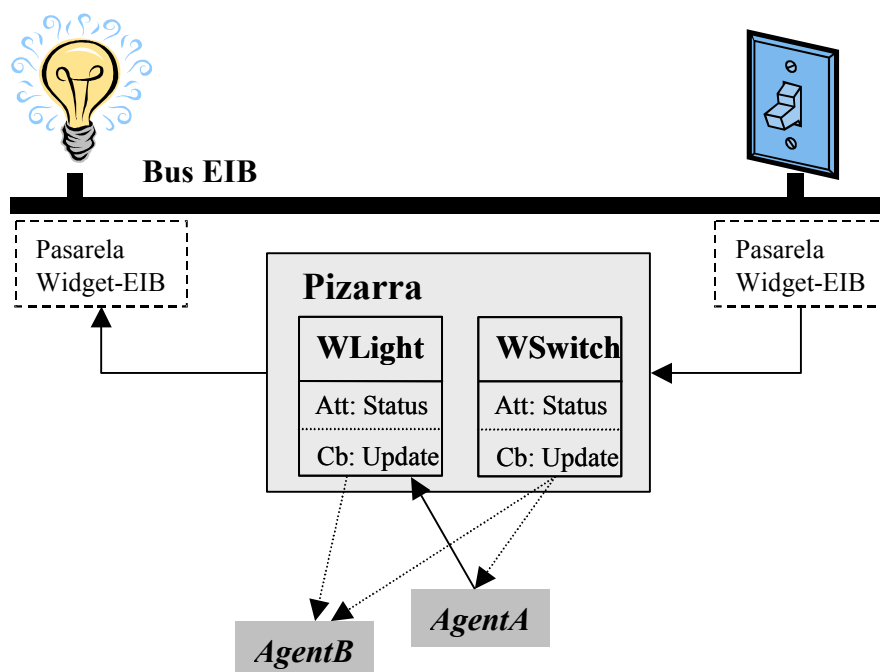


Figura 17. Ejemplo de pizarra utilizando la Context Toolkit

A continuación se presenta el código de la estructura de la pizarra, y de los dos widgets:

Pizarra

```
package odisea.arch.blackboard;

import odisea.arch.widgets.*;

import java.util.Hashtable;
import java.util.Vector;
import java.util.Enumeration;

import context.arch.widget.Widget;
import context.arch.util.ContextTypes;
import context.arch.comm.DataObject;
import context.arch.storage.Attribute;
import context.arch.storage.Attributes;
import context.arch.storage.AttributeNameValue;
import context.arch.storage.AttributeNameValues;

public class Blackboard {

    Hashtable bb;

    public Blackboard(){
        bb = new Hashtable();

        // Un widget se caracteriza por un puerto de comunicaciones, un identificador y por si
        // se quiere guardar la historia de cambios
        WLight w1 = new WLight(4444,"Light1",false);
        bb.put("Light1",w1);
        System.out.println("Cargado Light1");

        WSwitch w2 = new WSwitch(4445,"Switch1",false);
        bb.put("Switch1",w2);
        System.out.println("Cargado Switch1");
    }
    public static void main(String args[]){
        Blackboard bbt = new Blackboard();
    }
}
```

WLight

```

import context.arch.comm.DataObject;
import context.arch.subscriber.Callbacks;
import context.arch.service.Services;
import context.arch.storage.Attributes;
import context.arch.storage.Attribute;
import context.arch.storage.AttributeNameValues;
import context.arch.util.ContextTypes;
import context.arch.widget.Widget;
import context.arch.util.Constants;

public class WLight extends Widget {

    public static final String CLASSNAME = "Light";
    public static final String VERSION_NUMBER = "0.0.1";

    // Attributes
    public static final String STATUS = "status";

    public WLight (int port, String id, boolean storageFlag){
        super(port,id,storageFlag);
        setVersion(VERSION_NUMBER);
        // setup
        AttributeNameValues initialValues = new AttributeNameValues();
        initialValues.addAttributeNameValue(STATUS, new Integer(0), Attribute.INT);
        store(initialValues);
    }
    protected Attributes setAttributes(){
        Attributes atts = new Attributes();
        atts.addAttribute(STATUS,Attribute.INT);
        return atts;
    }
    protected Callbacks setCallbacks(){
        Callbacks calls = new Callbacks();
        calls.addCallback(UPDATE,setAttributes());
        return calls;
    }
    protected AttributeNameValues queryGenerator(){
        return new AttributeNameValues();
    }
}

```

WSwitch

```
package odisea.arch.widgets;

import context.arch.comm.DataObject;
import context.arch.subscriber.Callbacks;
import context.arch.service.Services;
import context.arch.storage.Attributes;
import context.arch.storage.Attribute;
import context.arch.storage.AttributeNameValues;
import context.arch.util.ContextTypes;
import context.arch.widget.Widget;
import context.arch.util.Constants;

public class WSwitch extends Widget {

    public static final String CLASSNAME = "Switch";
    public String VERSION_NUMBER = "0.0.1";

    public static final String STATUS = "status";

    public WSwitch (int port, String id, boolean storageFlag){
        super(port,id,storageFlag);
        setVersion(VERSION_NUMBER);
        // setup
        AttributeNameValues initialValues = new AttributeNameValues();
        initialValues.addAttributeNameValue(STATUS, new Integer(0), Attribute.INT);
        store(initialValues);
    }
    protected Attributes setAttributes(){
        Attributes atts = new Attributes();
        atts.addAttribute(STATUS,Attribute.INT);
        return atts;
    }
    protected Callbacks setCallbacks(){
        Callbacks calls = new Callbacks();
        calls.addCallback(UPDATE,setAttributes());
        return calls;
    }
    protected AttributeNameValues queryGenerator(){
        return new AttributeNameValues();
    }
}
```


Apéndice II.

Ejemplo de la Context Toolkit utilizando el lenguaje XML y el protocolo HTTP

Se muestran dos ejemplos de cómo la Context Toolkit utiliza el protocolo HTTP y el lenguaje de representación XML. El primer mensaje sirve para que una aplicación se suscriba a los cambios en los atributos de un widget. Las suscripciones se identifican por *callback*: cada *callback* se caracteriza por un identificador y un conjunto de atributos a los que se suscribe.

El mensaje tiene dos partes:

- Cabecera: ésta es la parte del protocolo HTTP. La primera línea indica que se está utilizando el método POST, y que se está accediendo al URI `addSubscriber`. La Context Toolkit interpreta esta dirección como el método al que tiene que pasarle el cuerpo del mensaje. El resto son parámetros del protocolo HTTP, como el tipo de contenido o la longitud en bytes del cuerpo.
- Cuerpo: está constituido por el contenido XML. El significado de los campos es:
 - Identificador del widget al que se está suscribiendo, *PersonNamePresence2_test*
 - Identificador de la aplicación subscriptora: *SimpleApp*
 - Dirección de la máquina donde se encuentra *SimpleApp*
 - Puerto de escucha de la aplicación *SimpleApp*.
 - Nombre del *callback* a la que se suscribe: *update*
 - Etiqueta que tiene que devolver el widget junto con el cambio que se ha producido: *presenceUpdate*
 - Las dos últimas etiquetas sólo se utilizan si se ponen condiciones de cuándo se quiere que haya aviso de los cambios en los atributos.

Mensaje para suscribirse a un evento

```
POST addSubscriber HTTP/1.0
```

```
User-Agent: Context Client
```

```
Host: cc115227-a
```

```
Content-Type: text/xml
```

```
Content-Length: 349
```

```
<?xml version="1.0"?>
<addSubscriber><id>PersonNamePresence2_test</id>
<subscriber><subscriberId>SimpleApp</subscriberId>
<hostname>24.5.188.61</hostname>
<port>9999</port>
<callbackName>update</callbackName>
<callbackTag>presenceUpdate</callbackTag>
<conditions></conditions>
<attributes></attributes>
</subscriber>
</addSubscriber>
```

El siguiente mensaje es la respuesta del widget a la petición del suscriptor. En este caso devuelve la identificación del suscriptor y que no ha habido error.

Respuesta al mensaje anterior

```
HTTP/1.0 200 OK
Date: Sat Feb 03 18:35:51 EST 2001
Server: context/1.0
Content-type: text/xml
Content-length: 136
```

```
<?xml version="1.0"?>
<subscriptionReply><subscriberId>SimpleApp</subscriberId>
<errorCode>noError</errorCode>
</subscriptionReply>
```