Universidad Autónoma de Madrid

TESIS DOCTORAL

Estudio e Integración de un Sistema de Diálogos Dinámico en un Entorno Inteligente

AUTOR
Germán Montoro Manrique

DIRECTOR
Xavier Alamán Roldán

PROGRAMA DE DOCTORADO
Ingeniería Informática y de Telecomunicación
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma De Madrid

Escuela Politécnica Superior

Febrero de 2005

Índice

Agradecimientos

"De gente bien nacida es agradecer los beneficios que reciben"
Don Quijote – Capítulo XXII – Primera Parte

A mis padres, José y Rosario, maestros, a los que tanto admiro, de los que tanto aprendo. Con su amor y sabiduría sin límites, que han hecho posible que yo escribera esta tesis, que yo sea quien soy. Podría decir *Todo sobre mis padres*, y nunca sería suficiente.

A Isabel, *La fiera de mi niña* y *La niña de mis ojos*. La mejor de las dichas que se ha podido cruzar en mi vida. La sonrisa que aflora permanentemente en mis labios. El mayor de mis soportes, quien me hace ver que ha merecido la pena.

A Pablo, maravillosamente único. Si alguna vez hubiera podido desear al amigo y compañero, jamás mi imaginación hubiera llegado tan lejos. Cuánto te debo de lo que hay en esta *Tesis*, cuánto de mi forma de ver el mundo. Una *Quimera de oro* hecha realidad.

A Xavier, que me ha enseñado tantas y tantas cosas en estos años y que obstinadamente ha confiado en mí. *El Padrino* que me ha ofrecido *Alas de mariposa* para llegar más allá de lo que mis ojos veían.

A Alejandro, el mejor bálsamo para los *Tiempos modernos*. Amigo y confidente incansable. La generosidad de un afecto que me ha dado tanto y que constituye una de mis más valiosas pertenencias. A Estrella, un *Ángel azul* que me hizo cruzar el mar para recibirme, como siempre que lo he necesitado, con el regalo de su sonrisa. A Pedro, un diamante en bruto que no debe faltar en un *Atraco perfecto*. Una suerte para quienes podemos disfrutar de su amistad silenciosa y sin reparos. A Miguel, compañero de penurias, de *Sonsiras y lágrimas*. A Ana, que te proporciona el placer de *La gran evasión*. A Rosa, de quien aprendes que *La vida es bella*. A Enrique, *El hombre tranquilo*, que te premia con su saber en forma de bondad. A Juana, *Poderosa Afrodita*, una ayuda inestimable que transforma las dificultades en deleite. A *La cuadrilla*, con quienes he pasado y paso grandes momentos: Leila, Manu, Fran, Abraham, José, Carlos, Ruth, Abdel, Álvaro. A todos los que han hecho posible esta *Odisea en el espacio*: José, María, Carlos, Rubén y tantos otros.

A quienes han confiado en mí, para este y otros trabajos, mucho más allá de lo que nunca me hubiera atrevido a aventurar. Que me han enseñado a *Tener y no tener*, a aprender de los *Acordes y desacuerdos*. A *Los olvidados* que, por omisión, no he mencionado pero saben que tienen que estar aquí.

A Miguel de Cervantes Saavedra, su *Ingenioso hidalgo don Quijote de la Mancha* y su extraordinario escudero Sancho Panza.

A Javier Martínez, en nuestra memoria, con nuestro agradecimiento.

1 Motivación

"Y en lo de forzarles que estudien esta o aquella ciencia no lo tengo por acertado, aunque el persuadirles no será dañoso"
Don Quijote – Capítulo XVI – Segunda Parte

Durante los últimos años ha surgido dentro de la comunidad científica que trabaja en el ámbito de las interfaces de usuario (Human Computer Interfaces) un nuevo área de trabajo que está recibiendo una gran atención: los llamados entornos inteligentes (intelligent environments o smart environments). Últimamente también ha recibido el nombre de inteligencia ambiental (ambient intelligence).

Los entornos inteligentes interactúan con el individuo y le ayudan de forma natural en la realización de las tareas cotidianas. Los ordenadores dentro de un entorno inteligente suelen quedar ocultos para el usuario, los servicios del sistema se obtienen mediante una interacción sensible al contexto. De este modo, se consigue que las habitaciones, oficinas y otros espacios habitables tengan una entidad propia, puedan tomar la iniciativa en la interacción y mejoren la calidad de vida de sus ocupantes.

Este tipo de sistemas proporciona "entornos altamente interactivos que usan computación embebida para observar y participar en los asuntos cotidianos del mundo que les rodea" [Coen, M.H. 1998]. Muchos de estos entornos inteligentes pretenden cambiar el significado clásico del uso de un ordenador. Acoplando los ordenadores a los entornos cotidianos las personas pueden interactuar con ellos de la misma manera que lo hacen con el resto de la gente: mediante el habla, los gestos, el movimiento y/o el contexto.

Los entornos inteligentes se encuentran ante el reto de llenar los espacios de servicios computacionales sin que supongan una intromisión al usuario. Las personas deben poder utilizar estas nuevas utilidades de forma natural y sencilla. Estos entornos, en todo caso, deben suponer una mejora en la calidad de vida para las personas que los habitan.

Sus aplicaciones se encuentran en el campo del hogar, de la oficina y se expanden hasta englobar cualquier tipo de entorno en el que se puedan encontrar las personas. Los usuarios a los que van dirigidos abarcan el más amplio abanico. Su aceptación pasa porque un usuario, en cualquier momento, pueda encontrarse dentro de un entorno inteligente sin tener que apreciar diferencias sustanciales, o al menos sin que suponga una desventaja, con respecto a entornos convencionales. Un campo donde su aplicación es de especial interés es el de las personas mayores, con discapacidades o enfermedades.

Estos sistemas ya han iniciado su desarrollo, habiéndose conseguido entornos interactivos con diferentes capacidades que han llevado a la práctica los estudios teóricos. Su paulatino perfeccionamiento, junto con el estudio de nuevos métodos de interacción natural, hará que su implantación se extienda y que la sociedad los acepte como sistemas que pueden suponer una gran ayuda en el desarrollo de las actividades cotidianas [Coen, M.H. 1999].

La presencia de la tecnología en los hogares y en los entornos de convivencia, que ha crecido de forma continuada en las últimas décadas, constituye una innegable mejora en la calidad de vida. Desde los elementos más sencillos (despertadores digitales programables, termostatos que controlan la temperatura, contestadores que nos avisan

si ha llamado alguien, etc.) hasta aquellos que resultan más vitales (servicios de teleasistencia para personas mayores en caso de urgencia, detectores de intrusos, etc.). El siguiente paso en esta democratización de la tecnología consiste en extenderla hasta límites insospechados hace sólo unos años para crear una revolución tecnológica invisible en nuestro modo de vida.

Para que la irrupción de esta nueva tecnología ubicua en nuestros espacios sea una auténtica revolución es necesario un cambio en la forma de actuar, de modo que los entornos pasen a ser *inteligentes*. La inteligencia del entorno se manifiesta en dos cualidades no excluyentes que pueden estar presentes de forma conjunta o por separado:

- Por un lado, poseer la capacidad de predecir el comportamiento y las necesidades de sus ocupantes y de asistirles. Para esto será necesario que puedan conocer quiénes son sus ocupantes, qué tareas realizan, cuáles son sus preferencias, requerimientos especiales, etc. Algunas de las aplicaciones de esta aproximación en un ámbito general son el control automático de elementos del entorno (las luces si alguien entra, la calefacción si hace frío, etc.), regulación del consumo de energía (ya sea de forma automática o enseñando pautas de consumo), la seguridad en el entorno (detección de catástrofes o intrusos, prevención de accidentes, etc.), etc. En un campo de aplicación más específico pueden servir como apoyo a personas enfermas o ancianos (para controlar sus constantes vitales, supervisar un proceso médico, actuar en caso de necesidad sanitaria, etc.). En este caso el entorno actúa de forma automática, adelantándose a las necesidades de los usuarios, sin que resulte necesaria la interacción explícita con estos.

- Por otro lado, ayudar a sus ocupantes en la realización de las tareas cotidianas o a desenvolverse en el entorno. Para esto se han de crear modos de interacción naturales, haciendo que la tecnología *desaparezca*. Las aplicaciones de este punto de vista pueden facilitar una interacción más intuitiva (interfaces ubicuas y naturales para acceder a los elementos del entorno), simplificar el uso de la tecnología (las mismas interfaces permiten de forma natural encender una luz, programar un vídeo o enviar un mensaje, sin necesidad de conocimientos previos), crear un ambiente homogéneo entre espacios (las interfaces y la información puede acompañar a los individuos por los entornos), etc. De esta forma el entorno proporciona un modo de interacción natural: el usuario es capaz de dialogar con el sistema y éste responde y atiende sus solicitudes.

Dependiendo de las características con que se desee dotar al entorno algunas de estas aproximaciones se considerarán imprescindibles y otras superfluas. En la presente tesis se considera que uno de los elementos fundamentales para constituir un entorno inteligente completo es el sonido, como un modo de interacción natural e intuitivo con el entorno. Aunque, como se ha podido ver, la presencia de sonido no es en sí un requisito necesario para la existencia de un entorno inteligente, sí constituye una parte esencial para la construcción de un entorno de máxima interacción con sus habitantes en un modo natural. Los humanos consideramos al habla como una forma habitual,

espontánea y sencilla para comunicarnos entre nosotros [Clark, H.H. and Brennan, S.E. 1991]. Evidentemente, esta vía de comunicación puede ser sustituida por otras en aquellos contextos en que las circunstancias lo requieran. Además, este modo de comunicación puede no resultar el más eficiente en personas con deficiencias auditivas y/o del habla. Por lo tanto, aunque no siempre sea la mejor forma de entrada de información, sí supone un método poderoso para la creación de entornos de comunicación persona-ordenador [Karat, C. et al. 1999]. La interacción continua dentro de un espacio de convivencia cotidiana sin la posibilidad de utilizar la voz supondría un esfuerzo considerable para el habitante del entorno, lo que disminuiría de forma notable su capacidad de acción y le impediría desenvolverse en el mismo de un modo natural.

Hasta el momento se han explorado las posibilidades de las interfaces conversacionales orales en el escenario clásico del escritorio o en sistemas telefónicos de banca, reserva de billetes de tren, planificación de rutas, etc. Es necesario extender esta exploración al contexto de un entorno inteligente, en donde el foco de atención del usuario se desplaza sobre un conjunto dispar de dispositivos físicos conectados en red. Dentro del campo de los entornos inteligentes todavía no se han dedicado suficientes esfuerzos a investigar las posibles formas de interacción con los elementos que lo componen.

Aunque existe investigación relacionada con las interfaces en el hogar, su foco se ha centrado principalmente en tareas específicas para los ordenadores personales [Tetzlaff, L. et al. 1995], en lugar de en entornos inteligentes integrados. Sin embargo las asunciones implícitas de diseño para un ordenador personal son inapropiadas en otros entornos más genéricos [Mateas, M. et al. 1996].

Por otro lado, un estudio llevado a cabo mediante técnica de *Mago de Oz* (ver apartado 2.9.2) en el entorno inteligente EasyLiving (ver apartado 2.4.5), realizado con el propósito de conocer lo que esperan los usuarios de un entorno inteligente, demuestra que las personas prefieren utilizar la voz para controlar los dispositivos del hogar y que, cuando a los participantes se les proporciona la opción de elegir entre diversos métodos para controlar los elementos del entorno, éstos eligen su voz [Brumitt, B. and Cadiz, J.J. 2001].

Por lo tanto, una de las premisas de esta tesis es considerar los sistemas de diálogos orales como elementos esenciales para el desarrollo completo de un entorno inteligente. Evidentemente, las necesidades y características de estos sistemas de diálogos difieren en algunos puntos a las de otros sistemas implementados. Uno de los aspectos que se han considerado fundamentales viene definido por la idiosincrasia de los entornos inteligentes. Estos son espacios altamente dinámicos cuya configuración puede cambiar de forma considerable. Los dispositivos se pueden añadir o eliminar del entorno, los usuarios entran y salen del mismo y éstos pueden traer consigo nuevos dispositivos móviles que se integren en el mismo. Cualquier interfaz o modo de interacción debería ser capaz de adaptarse a esta dinámica permitiendo nuevas interacciones cuando sea necesario y obviando otras.

Esta adaptación de la interfaz al entorno inteligente no debería ocurrir exclusivamente para una interfaz de diálogos orales sino que debería poder generalizarse fácilmente para cualquier otro tipo de interfaz con que se desee dotar al entorno. Este es otro de los aspectos considerados en la presente tesis. Mediante un lenguaje de descripción (DL, por sus siglas en inglés) del entorno se puede especificar, además de la composición del entorno y sus elementos, las interfaces que se desean crear para el mismo.

Centrándose en la interfaz oral, el sistema de diálogos y las posibles interacciones deben ser distintos para entornos diferentes y estos cambios se deben poder realizar con un esfuerzo mínimo o nulo por parte de los diseñadores de la interfaz de diálogos orales. Por lo tanto en esta tesis se propone el estudio e integración de un sistema de diálogos que se crea automáticamente adaptándose a las características del entorno inteligente.

En esta tesis se presenta una propuesta de integración de un sistema de diálogos orales de creación automática con un entorno inteligente. Para ello, en el capítulo 2 se analiza en primer lugar el estado del arte de los entornos inteligentes y de los sistemas de diálogos, tanto de forma aislada como en conjunto. A continuación, en el capítulo 3, se desarrolla una propuesta de entorno inteligente, analizando sus componentes y la forma de definirlos. En el capítulo 4 se muestra un primer sistema de diálogos desarrollado para interactuar con el entorno. Su implementación permitió diseñar e implementar un sistema de diálogos automático de más alto nivel, que se detalla con detenimiento en el capítulo 5. A continuación, en el capítulo 6, se muestran las pruebas de evaluación objetivas y subjetivas realizadas con usuarios de la interfaz en el entorno. Finalmente, el capítulo 7 concluye con una presentación de las aportaciones, publicaciones a las que se ha dado lugar y el trabajo futuro.

2 Estudio de las áreas de investigación relacionadas

"El que lee mucho y anda mucho, ve mucho y sabe mucho"
Don Quijote – Capítulo XXV – Segunda Parte

Aunque hace varias décadas que se trabaja en el área de las interfaces de usuario, hasta muy recientemente no han empezado a aparecer sistemas que se puedan considerar con propiedad entornos inteligentes. De hecho, la primera conferencia internacional dedicada exclusivamente a este tema tuvo lugar a finales de 1999 [MANSE 1999]. Actualmente los grupos más activos en el ámbito internacional en el desarrollo de entornos inteligentes son el Laboratorio de Inteligencia Artificial y el Media Lab, ambos del Massachusetts Institute of Technology (MIT), y el College of Computing de Georgia Tech. Estos tres departamentos son considerados entre los más punteros en investigación en áreas relacionadas con la interacción hombre-máquina. También realizan investigación dentro de este campo el Institute of Cognitive Science de la University of Colorado, el Vision Group de Microsoft Research, el Medical Automation Research Center de la University of Virginia o los Interactive Systems Laboratories de la Carnegie Mellon University.

En Europa también es muy reciente la aparición de trabajos en entornos inteligentes. Se pueden mencionar como grupos de investigación activos en este tema el Institut National de Recherche en Informatique et en Automatique (INRIA) francés, el Institut für Telematik de la alemana Universität Karlsruhe, el Centro Alemán de Investigación de Inteligencia Artificial (DFKI) o el Institut für Pervasive Computing austriaco. Un proyecto destacado en el campo de la industria es el Philips HomeLab. Por último, dentro del VI Programa Marco de la Unión Europea se han desarrollado acciones específicas orientadas a la potenciación de la investigación en el campo de la inteligencia ambiental (*ambient intelligence*).

En España, aunque existe una amplia tradición de investigación en cada una de las técnicas necesarias para la construcción de entornos inteligentes (visión artificial, reconocimiento y análisis del habla, arquitecturas de agentes heterogéneos, interfaces de usuario), son pocos los esfuerzos de integración de todas las anteriores técnicas para crear un entorno inteligente. En este sentido se puede mencionar las casas domóticas que están desarrollando Telefónica e Ikerlan y el proyecto Voice Smart Home de la Universidad Politécnica de Valencia.

La relevancia de los entornos inteligentes queda así mismo avalada por el VI Programa Marco de investigación científica de la Unión Europea en donde la inteligencia ambiental tiene una importancia primordial dentro del área de las tecnologías de la información.

2.1 Computación ubicua

Los entornos inteligentes se basan en el concepto de computación ubicua que fue inicialmente definido por [Weiser, M. 1991]. Para Weiser, la ubicuidad y la transparencia del sistema son las principales características de la computación ubicua. Existen dos aproximaciones principales para satisfacer el requerimiento de ubicuidad:

- Que el sistema sea lo suficientemente móvil como para poder ser transportado por el usuario.

- Que el sistema sea un espacio instrumentado que rodea al usuario en su posición, por ejemplo, una habitación, una oficina, un edificio.

De este modo se obtienen herramientas con dos características fundamentales [Weiser, M. 1994]:

- Son herramientas invisibles, esto es, no son perceptibles en la conciencia del usuario. Así las personas sólo se tienen que centrarse en la tarea que están acometiendo.
- Son herramientas con un comportamiento tan cercano como es posible al del ser humano.

Estas ideas, junto con numerosos trabajos experimentales, han propiciado un gran interés en la computación ubicua. Su reto consiste en aumentar la penetración de los ordenadores en el medio a la vez que disminuyen su intromisión en los entornos cotidianos y proporcionan servicios más valiosos a los usuarios. Con este tipo de sistemas el usuario se debe ahorrar el proceso de buscar y encontrar la interfaz con el ordenador para hacer que sea la interfaz quien tome la responsabilidad de localizar y servir al usuario [Abowd, G.D. 1998].

Todo este trabajo en computación ubicua tiene un gran impacto en la motivación e intereses de numerosos grupos de investigación para establecer una infraestructura de computación ubicua a gran escala. Desde su definición, el punto de vista aquí desarrollado permanece como una constante utilizada para la creación de entornos inteligentes [Nixon, P. et al. 1999].

2.2 Características de un entorno inteligente

A pesar de que los entornos inteligentes son todavía muy recientes ya existen varios trabajos que intentan acotar las características necesarias para su creación. Aunque los requisitos mínimos necesarios pueden variar dependiendo del punto de vista y la aproximación tomada, todos se basan en el concepto de computación ubicua como punto de partida para la creación de un entorno con características computacionales avanzadas al servicio de los usuarios.

Según [Pentland, A. 1996] un requisito necesario es que los ordenadores sean capaces de ver y oír lo que la gente hace para que así puedan llegar a ser realmente útiles. Las habitaciones inteligentes han de poseer cámaras y micrófonos que transmitan su información a una red de ordenadores. Gracias a esta conexión los usuarios pueden usar sus acciones, voces y expresiones para interaccionar con el ordenador. A su vez, el entorno ha de ser capaz de conocer:

- Dónde se encuentran los usuarios dentro del entorno.
- Quiénes son los usuarios que se encuentran en el entorno, qué están diciendo y a quién se están dirigiendo.

- Qué están haciendo los usuarios en el entorno. Debe analizar las expresiones faciales y el habla siempre dentro del contexto en el que se encuentran.
- Por qué el usuario está realizando las acciones, de modo que el sistema pueda reaccionar ante ellas.
- Combinando estas capacidades se pueden construir entornos inteligentes en los que los ordenadores son capaces de comunicarse con los usuarios de forma no obstructiva y el entorno puede incluso percibir estados de atención, emoción y adaptarse a ellos. [Nixon, P. et al. 1999], a las características hasta ahora descritas como necesarias en la construcción de un entorno inteligente, añade una nueva característica deseable, el aprendizaje: los entornos deben poder aprender patrones de actuación para mejorar la calidad de la interacción entre el sistema y los usuarios.

Sin embargo no todos los trabajos presentan el mismo punto de vista. Basándose también en este concepto de computación ubicua, [Shafer, S.A.N. 1999] se aleja de la idea hasta ahora comentada de computación invisible (en la que el usuario puede llegar a ignorar completamente la existencia de un ordenador) para proponer un conjunto de interacciones explícitas e implícitas que se pueden combinar para conseguir un mayor aprovechamiento de los recursos disponibles. Propone como medida necesaria para la creación de estos espacios unas características fundamentales que todos los entornos inteligentes deben compartir:

- Los sistemas deben poder permitir que los diferentes dispositivos de los que se componen se comuniquen entre ellos proporcionándose información significativa. Algunas de estas interacciones se han de producir de forma automática pero el usuario siempre debe poder ser consciente de cómo se producen. Por lo tanto es necesario un descubrimiento automático de dispositivos y una adaptación automática del comportamiento basada en la configuración y el estado de los dispositivos del sistema.
- Los entornos inteligentes han de poseer un conocimiento del mundo. Esto requiere algún modelo de representación para comunicar las percepciones de los sensores a las aplicaciones y a otros elementos del sistema, esto es, se necesita una ontología de los conocimientos.
- Se debe tener en cuenta diferentes aspectos para modelar a la gente. El sistema debe monitorizar las actividades de los usuarios y utilizar parte de esta información para mejorar su rendimiento.
- Se ha de determinar dónde van a residir los datos generados por el sistema y cómo se van a almacenar, si es necesario que la información migre de un lugar a otro y cómo integrar información de elementos estáticos y dinámicos.
- Los sistemas necesitan una ontología que estandarice las descripciones de los recursos a la vez que un mecanismo de nombres persistente. Se necesita disponer de mecanismos que permitan o denieguen accesos a cada uno de los recursos.

- Los entornos inteligentes tienen que permitir el crecimiento y la mejora de cada una de las partes que los constituyen. Se ha de permitir añadir nuevos dispositivos o conjuntos de dispositivos que se puedan utilizar de forma inmediata. Además, la ontología utilizada y las aplicaciones tienen que poder adaptarse a las nuevas necesidades.
- Deben definir un paradigma de interfaz entre los programas y los dispositivos que permita que el software se adapte al hardware del sistema.
- El entorno tiene que disponer de mecanismos de control automático del comportamiento del sistema aunque también han de permitir un control directo por parte de los usuarios. El sistema debe tener almacenadas especificaciones de comportamiento que determinen bajo qué condiciones se deben llevar a cabo ciertas acciones.
- El entorno ha de mejorar la vida de los usuarios que lo utilizan. Se tiene que determinar quién toma el control del proceso en cada momento y qué grado de libertad se proporciona a los usuarios.
- Un entorno ubicuo debería ser único para todo el mundo aunque tenga numerosas variaciones locales entre diferentes lugares. Este entorno debería estar evolucionando continuamente.

Un nuevo conjunto de características es definido por [Johanson, B. et al. 2002], que complementa los mostrados anteriormente:

- Múltiples dispositivos pueden estar presentes de forma simultánea en el entorno y cada uno se mostrará más eficaz en realizar una tarea en particular y/o presentar la información.
- También existirá un conjunto heterogéneo de aplicaciones software ejecutándose en estos dispositivos, incluyendo aplicaciones comerciales y ad-hoc. Se debe poder acceder a todas estas aplicaciones de forma estándar de modo que los usuarios puedan tratarlas como un conjunto homogéneo.
- Las interfaces deben ser adaptadas a diferentes modalidades de entrada y salida, como voz o vídeo.
- El entorno puede tener a múltiples usuarios, dispositivos y aplicaciones activos a la vez. Los dispositivos podrían apagarse, dispositivos inalámbricos podrían entrar o salir del entorno o parte del equipo podría dejar de funcionar en periodos transitorios. Los entornos evolucionan a lo largo del tiempo, cambiando su configuración. Por lo tanto el entorno debería ser capaz de adaptarse de forma dinámica a estos cambios o, incluso, adelantarse a ellos.

Los entornos inteligentes han evolucionado usando algunas de las características aquí expresadas para intentar satisfacer las nuevas necesidades que permiten hacer realidad el paradigma de la computación ubicua.

2.3 Avances en la investigación en entornos inteligentes

Los esfuerzos pioneros se basaron en una infraestructura fija para proporcionar servicios computacionales al usuario. Un ejemplo de tales entornos es el Olivetti Active Badge system [Want, R. et al. 1992], que cuenta con una infraestructura de balizas por infrarrojos instaladas en cada oficina y habitación a lo largo de un edificio. Los usuarios llevan un identificador (*badge*) personal que les identifica y comunica su posición al servidor central. Algunas de las aplicaciones iniciales de esta tecnología consistieron en localizar rápidamente a una persona en un campus o teletransportar automáticamente su ordenador personal a la estación de trabajo más cercana [Rosu, M. et al. 1997].

El Xerox ParcTab system [Want, R. et al. 1995] va un paso más adelante al dotar al usuario con la capacidad de computación móvil. ParcTab se basa en un sistema de localizadores personales conscientes de la posición y equipados con una pantalla monocroma y una interfaz basada en bolígrafo. Mediante una infraestructura de comunicación se proporciona acceso remoto a las estaciones de trabajo o servidores de datos, visionado remoto de ficheros, diccionario, acceso a la Web, etc.

Una extensión más reciente a los sistemas de Active Badges es el sistema Audio Aura, en el que los usuarios llevan un localizador personal y auriculares inalámbricos [Mynatt, E.D. et al. 1997]. Cuando se mueven por el edificio, transmisores de radio frecuencia proporcionan información personalizada de la posición del usuario en forma de una señal de audio. Otras extensiones incluyen el sistema Palplates, que consiste en pequeños visualizadores (*displays*) sensibles al tacto instalados en posiciones estratégicas del entorno de una oficina: en salas de reuniones, en la cocina y junto a las impresoras [Mankoff, J. and Schilit, B. 1997]. Más recientemente, esta tecnología se ha visto ampliada con el uso de RFID (Radio Frequency Identification), un método para almacenar y recuperar información utilizando etiquetas RFID.

En todos los sistemas anteriormente descritos, los servicios computacionales están disponibles dondequiera que el usuario esté. Pero el acceso a estos servicios es apenas transparente: los usuarios tienen que llevar u operar con dispositivos. Entre los mencionados, Audio Aura es el que presenta más transparencia, excepto por el hecho de que los usuarios tienen que llevar auriculares que les aíslen del resto del mundo. Pueden encontrarse tendencias similares, que se concentran sin embargo mucho más en la computación embebida y en la inteligencia y receptividad en los dispositivos, dentro del proyecto Things that Think del MIT Media Laboratory [http://www.media.mit.edu/ttt].

En este mismo sentido avanza el área de trabajo conocida como realidad enriquecida (como posible traducción al español de la denominada *augmented reality)*. Por ejemplo, el Digital Desk de [Wellner, P. 1993] reemplaza la habitual metáfora del escritorio basado en la pantalla por uno real. Una cámara instalada en el techo y un proyector permiten que se utilice la superficie del escritorio como un dispositivo de entrada/salida. El usuario puede dibujar en una hoja de papel y beneficiarse de las ventajas de servicios computacionales, tales como las funciones de copiar/pegar (una

línea dibujada puede ser copiada y entonces pegada: las copias son proyectadas). El usuario también puede empezar con un documento digital ya existente (por ejemplo, proyectado) y utilizar herramientas del mundo real como el bolígrafo, el borrador o incluso las manos para manipular los datos. El escritorio en el mundo real es *mejorado* con capacidades computacionales. El prototipo Ariel de [Mackay, W.E. et al. 1995] traslada esta técnica de interacción al campo de ingenieros móviles que anotan dibujos de ingeniería en la construcción. En este sentido se orienta la investigación dentro del campo de las interfaces tangibles. Algunos proyectos desarrollados siguiendo este paradigma son [Ishii, H. and Ullmer, B. 1999]: *metaDESK, transBOARD* y *ambienROOM*, todos dentro de un proyecto denominado *Tangible bits* desarrollado en el Tangible Media Group del Media Lab. del MIT.

Existen otros prototipos de realidad "enriquecida" que requieren que el usuario lleve dispositivos especiales tales como visualizadores transparentes que proyectan anotaciones en una vista del mundo real. Sin embargo, los usuarios pueden moverse libremente por un espacio (por ejemplo, una biblioteca, una sala de mantenimiento). El sistema Karma para el mantenimiento de copiadoras [Feiner, S. et al. 1993] y el sistema Boeing's wearable para el mantenimiento de aviones [Sims, D. 1994] son representativos de esta aproximación. Extensiones a este tipo de trabajo han dado como resultado un nuevo campo de investigación titulado *Wearable Computing* (ordenadores que se llevan puestos). Los intereses de la investigación en esta área se dirigen hacia la potenciación de las capacidades humanas insertando la parte computacional en la ropa o en artefactos de uso diario.

En una vertiente similar, la investigación sobre automatización del hogar se ha enfocado en ocultar los dispositivos computacionales y en proporcionar una interacción transparente para poder acomodarse a usuarios sin conocimientos técnicos. Los esfuerzos en investigación de Microsoft Research o de IBM T.J. Watson Research Center son representativos de esta tendencia. Uno de los primeros esfuerzos en la investigación sobre un hogar con interacción transparente con el entorno de control es The Adaptive House [Mozer, M.C. 1998] (ver apartado 2.4.3). Equipada con sensores que detectan la presencia, el movimiento y las acciones de la gente, el sistema regula las luces y la calefacción en una casa de acuerdo con unos patrones de uso. Del mismo modo, la interacción transparente también ha sido examinada para entornos de oficina [Stafford-Fraser, Q. and Robinson, P. 1996]. Un buen ejemplo ya desarrollado de interacción transparente en una sala de conferencias es The Reactive Room de [Cooperstock, J.R. et al. 1995] (ver apartado 2.4.7.4).

Más recientemente, se ha generado un gran interés en profundizar en estas direcciones para conseguir interfaces naturales. Las técnicas de visión y audición computacional se emplean para proporcionar entornos inteligentes que "conozcan" a los usuarios. Los sistemas de visión se utilizan para localizar usuarios en un entorno, reconocerlos, y seguir sus gestos, expresiones y posturas corporales. The Reactive Room de [Cooperstock, J.R. et al. 1995] también usa técnicas de visión computacional para seguir la pista a los usuarios, al igual que los sistemas Smart Rooms [Pentland, A.

1996] y KidsRoom [Bobick, A. et al. 1997] del MIT Media Laboratory (ver apartado 2.4.7.3).

Otros ejemplos de entornos inteligentes e interactivos se encuentran en el MIT Artificial Intelligence Laboratory: HAL [Coen, M.H. 1998] (ver apartado 2.4.1), el MIT Media Laboratory: House_n [Intille, S.S. and Larson, K. 2003] (ver apartado 2.4.4) y el Future Computing Environments Group de Georgia Tech: Classroom 2000 [Abowd, G.D. et al. 1996] (ver apartado 2.4.7.5) y The Aware Home [Kidd, C.D. et al. 1999] (ver apartado 2.4.2).

2.4 Proyectos de investigación sobre entornos inteligentes

Aunque el número de proyectos todavía no es muy elevado, cada vez existen en el campo de los entornos inteligentes más trabajos que, basándose en las ideas de [Weiser, M. 1991], pretenden crear entornos con características de computación ubicua. Aunque casi todos ellos se basan en el uso transparente de nuevas tecnologías para mejorar la actividad cotidiana, entre ellos presentan analogías y diferencias como se puede ver en los siguientes apartados.

2.4.1 Intelligent room, HAL y Aire

HAL es un entorno altamente interactivo que utiliza computación embebida para observar y participar en los actos cotidianos que ocurren a su alrededor con el objetivo de crear un entorno inteligente. Es un descendiente de la Intelligent Room que ha continuado con los proyectos desarrollados y ampliado sus ideas. En su última evolución el sistema se denomina Aire y pretende estudiar y crear diferentes modalidades de entornos inteligentes. Todos estos sistemas se han desarrollado dentro del MIT Artificial Intelligence Laboratory.

2.4.1.1 Desarrollo inicial: Intelligent Room

La Intelligent Room se empezó a desarrollar en 1994 con el objetivo de explorar requisitos avanzados de interacción hombre-máquina y tecnologías de colaboración. Para ello crearon una habitación inteligente capaz de interpretar y aumentar sus actividades. Los tres aspectos fundamentales de este proyecto son [Torrance, M.C. 1995]:

- La habitación incorpora capacidades de presentación y de percepción que permiten interactuar y entender a sus ocupantes humanos. El sistema puede realizar seguimiento visual de los ocupantes, reconocimiento de gestos, mostrar información personalizada proyectada sobre las paredes de la habitación, etc.
- Dispone de un conjunto de agentes inteligentes que navegan por la Web de forma autónoma para recabar información o proporcionar servicios computacionales a la habitación de manera imperceptible.

- La habitación posee un conjunto de sistemas inteligentes que se combinan para dirigir las interacciones de la habitación, la gente y la Web manteniendo un repositorio de los datos y de los planes en una base de datos persistente.

El sistema es capaz de realizar seguimiento visual de hasta cuatro ocupantes de la habitación. Puede determinar si una persona está sentada, de pie, andando o señalando. Reconoce si varias personas están conversando, abrazándose o dándose la mano. Puede advertir si se está produciendo una reunión o si se está realizando una presentación.

Las personas y la habitación se pueden comunicar mediante el habla. Para ello la habitación incorpora un reconocedor de habla continua independiente del hablante y un sintetizador de voz.

Los usuarios pueden apuntar a elementos simbólicos de imágenes proyectadas en las paredes para seleccionarlos o referirse a ellos.

Para conseguir estos requisitos desarrollaron una serie de tecnologías en robótica, visión por ordenador, comprensión del lenguaje natural, procesamiento inteligente de la información y agentes autónomos (el sistema SodaBot [Coen, M.H. 1997]) que forman las características iniciales en el desarrollo de HAL.

2.4.1.2 Sistema posterior: HAL

Siguiendo con las ideas introducidas en la Intelligent Room, la motivación de este sistema es introducir los ordenadores en el mundo físico para apoyar lo que tradicionalmente se considera actividad no computacional, mejorando las actividades cotidianas de los usuarios [Coen, M.H. 1998]. Para ello se ha de permitir a los ordenadores participar en tareas no tratadas hasta ese momento de forma computacional y a los usuarios interactuar con los ordenadores de la misma forma que lo harían con otras personas, esto es, con los gestos, la voz, el movimiento y el contexto. Al tratarse de un sistema que utiliza diversos modos de interacción, está equipado con numerosos sistemas de visión computacional y de reconocimiento del lenguaje oral y gestual.

El sistema es capaz de observar dónde se mueve la gente, dónde señalan (bajo ciertas circunstancias) y escuchar y reconocer un conjunto variado de oraciones. Estas tareas se consiguen realizando mínimas modificaciones en el entorno y sin utilizar elementos adicionales ni tener que portar ningún dispositivo (en contraposición a otros sistemas [Want, R. et al. 1992]). Estas ideas se basan en dos principios fundamentales:

- Los ordenadores han de ser fáciles de utilizar y esencialmente invisibles a los usuarios.
- La información visual de una cámara es capaz de proporcionar más información que tecnologías sensitivas más simples.

Aunque el sistema ha sido probado dentro del entorno de la oficina, se ha construido de forma que se adapte fácilmente a cualquier tipo de entorno y que las interacciones multimodales no sean excesivamente específicas a sus aplicaciones (al contrario de lo

que ocurre en algunos sistemas demasiado ligados a un dominio, ver más adelante The KidsRoom o Classroom 2000). La arquitectura del entorno se basa en dos principios:

- El sistema se ajusta dinámicamente a la actividad de la habitación. Esto ocurre por ejemplo con el sistema de comprensión del habla [Coen, M.H. et al. 1999b]. En lugar de mantener una única gramática de reconocimiento activa, la habitación mantiene subconjuntos de pequeñas gramáticas activas que cambian dependiendo del contexto y las acciones del usuario.
- El sistema se construye de manera que resulte fácil de instalar y de mantener.

Para la implementación de la arquitectura del entorno se utiliza un modelo de agentes distribuidos denominada MetaGlue que constituye el pilar del proyecto. Según sus autores, esta estructura de agentes distribuidos es necesaria ya que [Coen, M.H. et al. 1999a]:

- La infraestructura de los entornos inteligentes tiende a ser altamente distribuida, reflejando la naturaleza distribuida del mundo real y la necesidad de estos entornos de grandes cantidades recursos computacionales.
- Los entornos inteligentes tienden a ser extremadamente dinámicos y requieren volver a configurar y gestionar de recursos sobre la marcha según cambian sus componentes, sus habitantes y las preferencias de los usuarios.
- Al estar compuestos de interfaces multimodales presentan altos grados de paralelismo para resolver múltiples sucesos simultáneos.
- La depuración de estos entornos presenta nuevos retos a sus creadores debido a su paralelismo y la dificultad para concretar su estado en un sentido computacional.

2.4.1.3 Sistema actual: Aire

El proyecto ha seguido creciendo y en la actualidad continúa con el nombre de Aire (Agent-Based Intelligent Reactive Environments). Hasta el momento han desarrollado diversos entornos inteligentes que engloban desde ordenadores de bolsillo a salas de reuniones. Estos trabajos forman parte fundamental del proyecto Oxygen [http://oxygen.lcs.mit.edu/], relacionado con la computación ubicua centrada en las personas.

Algunos de los proyectos desarrollados, además de la habitación inteligente mencionada anteriormente, incluyen:

- Intelligent Workspaces [Hanssens, N. et al. 2002]. Su propósito es construir entornos laborales que ayuden a sus ocupantes en las situaciones rutinarias de trabajo. El sistema se centra en crear espacios de trabajo dinámicamente reconfigurables y estudiar cómo la gente utiliza estos entornos laborales inteligentes.
- Ki/o. Este proyecto pretende transformar los espacios de encuentro públicos, tales como vestíbulos, zonas de descanso o ascensores en entornos inteligentes. En este

caso las capacidades computacionales y de percepción se sitúan en las paredes de estos entornos.

2.4.2 Domisilica y The Aware Home

Domisilica y The Aware Home son dos proyectos desarrollados en el Graphics, Visualization and Usability Center del Georgia Institute of Technology que buscan la creación de entornos inteligentes dentro del hogar.

2.4.2.1 Domisilica

Domisilica es un proyecto finalizado que se centra en la construcción de la casa del futuro. Si bien el proyecto HAL ha sido siempre probado en un entorno de oficina, Domisilica proponía un sistema enfocado en explorar los entornos inteligentes centrados alrededor del hogar [Abowd, G.D. et al. 1997]. Se planteaba la casa como una interfaz a la información y a los objetos asociados con ella. A esta información se podía acceder desde cualquier ordenador.

El proyecto se centraba fundamentalmente en tres puntos [Mankoff, J. et al. 1998]:

- Añadir capacidades computacionales a los objetos del mundo real localizados en la casa (al contrario de la aproximación de HAL).
- Proporcionar un modelo centralizado de la casa. Una base de datos en la que se almacena un modelo de la casa e información que no tiene correspondencia con el mundo real. Este modelo se utilizaba para integrar una amplia variedad de servicios (también en contraposición a la idea de agentes distribuidos de HAL).
- Hacer que la información fuera accesible de forma remota.

Se eligió la cocina como lugar para iniciar la investigación dentro de la computación en el hogar. Añadieron un monitor a la parte frontal del frigorífico (de hecho el proyecto se llamó en principio *CyberFridge*) en el que se mostraban notas virtuales y páginas Web "fijadas" en el frigorífico. Un usuario remoto podía ver o consultar por teléfono el contenido del frigorífico o cualquier información relacionada con el mismo.

Las actividades que llegó a admitir el sistema dentro del hogar fueron [Mankoff, J. and Abowd, G.D. 1997]:

- Actividades de comunicación. Entre ellas conversación o notas asíncronas.
- Gestión de inventario. Como por ejemplo mantener un seguimiento de los contenidos del frigorífico.
- Actividades de control. Tales como responder a la puerta o encender y apagar los electrodomésticos.

2.4.2.2 The Aware Home

The Aware Home [Kidd, C.D. et al. 1999] es un programa de investigación en curso basado en las ideas iniciales que cubría Domisilica. En él se ha construido una casa que sirve como un laboratorio viviente de computación ubicua y cotidiana en el hogar. El objetivo es crear la tecnología necesaria para que el entorno del hogar pueda percibir y proporcionar asistencia a sus ocupantes. El alcance de los proyectos que se desarrollan dentro de este programa comprende desde técnicas de desarrollo computacionales a estudios etnológicos y cognitivos.

El concepto de percepción ubicua busca desarrollar una infraestructura en red distribuida que soporte las actividades de forma transparente. Su interés se centra en pequeños sensores multimodales que proporcionen información espacio-temporal sobre el entorno (como ocurría en Domisilica y en oposición a HAL). Las características destacadas en el desarrollo de la percepción ubicua son [Essa, I.A. 1999]:

- Auto-calibración. Los sensores del entorno inteligente necesitan ser capaces de calibrarse automáticamente y adaptarse al entorno según las necesidades. Todos los sensores del entorno necesitan comunicar su estado y su área de cobertura al resto y desarrollar un modelo del entorno.
- Red. Se necesitan combinaciones de procesadores y sensores para construir entornos conscientes y elaborar una infraestructura de red. Una vez que se establece una red dinámica, los sensores pueden capturar de forma transparente flujos de información relevantes y compartirlos con otros sensores o procesadores.
- Computación distribuida. Para instalar todos estos servicios ubicuos es necesario estudiar y desarrollar una estructura computacional que los soporte. Esta infraestructura debe servir como cerebro del entorno donde se procesa toda la información que tiene que ver con el espacio.
- Sensores ópticos y de audio. Se utilizan los sensores de vídeo y audio como los sensores principales, capaces de interpretar las actividades que se realizan en el entorno.
- Sensores móviles. Que complementen a los micrófonos y cámaras estáticas. Estos sensores incluyen micrófonos o cámaras portadas por los usuarios o situados en plataformas móviles.
- Sensores embebidos. Pequeñas cámaras embebidas en el techo que permitan realizar una cobertura de todo el espacio y micrófonos embebidos en los muros y el techo que mejoren la tasa reconocimiento y permitan localizar el sonido.
- Otros sensores. Además de los sensores de audio y vídeo, otros tipos de sensores que "enriquecen" al usuario y al entorno. Estos van desde sensores de contacto que detectan qué mueble se está utilizando hasta una alfombra sensitiva capaz de hacer un seguimiento de las personas.

Con estos sensores ubicuos el espacio es capaz de aprender las actividades de los ocupantes y sus rutinas. Gracias a ellos el sistema puede tener diversos estados de consciencia del entorno [Essa, I.A. 1999]:

- Identidad. Se utilizan técnicas de identificación del hablante y reconocimiento de rostros como ayuda para obtener consciencia del entorno.
- Situación. Mediante seguimiento visual y táctil, y mediante el uso de un conjunto de micrófonos se localiza a los usuarios para obtener información sobre qué están haciendo.
- Actividad. Procesando el flujo de vídeo se sigue la pista de los usuarios al moverse obteniendo información sobre qué ocurre en un momento concreto o sobre la rutina de los usuarios.
- Seguimiento de la posición de los ojos. Se utiliza el reconocimiento de hacia dónde está mirando el usuario como señal para determinar qué está haciendo.
- Reconocimiento de gestos y expresiones. Las expresiones faciales ayudan a interpretar el estado afectivo del usuario y el reconocimiento de gestos ayudan en el análisis de los intentos comunicativos de los usuarios.
- Procesamiento del audio. Usando un conjunto de micrófonos se extrae una señal del habla más segura para los reconocedores del habla. Además realizan análisis no-verbal del habla para extraer información estilística codificada en el habla.

El trabajo dentro de The Aware Home se basa en cuatro temas complementarios:

- Diseño para la gente. Se diseñan experiencias interactivas apropiadas para la gente en un entorno del hogar consciente, centrándose en personas mayores.
- Tecnología. Se desarrolla tecnología sensitiva y de percepción altamente distribuida que permita conocer la actividad humana en los entornos físicos.
- Ingeniería del Software. Se afrontan los retos de construcción de pasarelas software robustas.
- Implicaciones sociales. Se exploran los beneficios y problemas políticos, sociales y legales relacionados con la privacidad y la autonomía cuando los servicios explotan la conciencia y el conocimiento de la actividad humana dentro del espacio protegido de un hogar.

2.4.3 The Adaptive House

The Adaptive House, desarrollada en el Institute of Cognitive Science de la University of Colorado, fue uno de los primeros proyectos en intentar desarrollar un entorno inteligente bajo el nombre inicial de The Neural Network House [Mozer, M.C. et al. 1995]. El sistema se centra en el control domótico de los componentes de una casa para dotarlos de inteligencia de modo que se adapten a sus habitantes (sin buscar la interacción natural con el entorno mediante el habla y/o los gestos, como los sistemas

vistos en los apartados anteriores). El objetivo esencial del proyecto es desarrollar una casa que se programe automáticamente observando el modo de vida y los deseos de sus inquilinos, y que aprenda para anticiparse y complacer sus necesidades.

Como ya se ha comentado, en lugar de permitir interaccionar con los usuarios de forma natural y realizar un seguimiento por medio de cámaras y micrófonos, The Adaptive House se centra en dos aspectos fundamentales [Mozer, M.C. 1998]:

- Regular sistemas básicos de control como la calefacción, la ventilación, el aire acondicionado, la temperatura del agua o la luz interior. El sistema monitoriza el entorno, observa las acciones realizadas por sus ocupantes e intenta deducir patrones en el entorno que predigan estas acciones. Si las acciones se pueden anticipar de forma segura el sistema puede llevarlas a cabo de forma automática, liberando a los ocupantes del control manual de la casa.
- Ahorrar recursos de energía cuando sea posible.

Para realizar estas acciones el sistema utiliza más de 75 sensores y actuadores:

- Sensores. Proporcionan información sobre la temperatura, los niveles de la luz de ambiente, el sonido, el movimiento y apertura de puertas y ventanas.
- Actuadores. Controlan el horno, los radiadores, el calentador del agua, las unidades de alumbrado, los ventiladores de techo, etc.

Los sistemas de control de la casa se basan en redes neuronales con aprendizaje por refuerzo y técnicas de predicción [Mozer, M.C. 1999]. Estos mecanismos permiten aprender de la experiencia y predecir cuándo se ha de encender la calefacción para producir un ambiente adecuado a la llegada de los ocupantes, detectar patrones estadísticos del uso del agua, deducir dónde se encuentran los habitantes y qué están haciendo para tomar medidas de control apropiadas o incluso predecir si se va a entrar en una habitación para encender las luces con antelación.

Los habitantes también pueden ajustar las condiciones del entorno manualmente (coincidiendo con una de las características expresadas como necesarias por [Shafer, S.A.N. 1999]). Esta información sirve como indicativo de que no se han satisfecho las necesidades de los usuarios y se utiliza como una nueva señal para el entrenamiento del sistema minimizando la necesidad de control manual hasta el mayor grado posible.

2.4.4 House_n

Se trata de la propuesta actual del consorcio formado por el Architecture y el Media Laboratory del MIT para el hogar del futuro. En este caso un equipo multidisciplinar está estudiando cómo crear entornos ubicuos para el hogar. Su objetivo es desarrollar tecnologías y diseñar estrategias que utilicen la información del contexto obtenida mediante sensores, facilitando la información a los usuarios en el lugar y el momento correctos [Intille, S.S. and Larson, K. 2003]. Como ocurre con los proyectos anteriores, para desarrollar el proyecto se ha construido un laboratorio que en este caso sirve como

infraestructura científica para estudiar el poder de la computación ubicua para motivar el cambio de comportamiento en el contexto del hogar.

El sistema utiliza la automatización de los dispositivos para ayudar a la gente a realizar las tareas que no puede realizar por sí misma debido a alguna discapacidad. Sin embargo, su principal objetivo no se centra en conseguir que la tecnología gestione de forma ubicua y activa los detalles del hogar. En este sentido se aleja de los proyectos hasta ahora mostrados, especialmente de The Adaptive House. Específicamente, los prototipos que construyen intentan demostrar cómo crear entornos que ayuden a las personas a vivir de forma autónoma en el hogar el mayor tiempo posible (en este caso coinciden en su interés en proporcionar asistencia a las personas mayores con The Aware Home), reducir el consumo de recursos e integrar el aprendizaje en las actividades diarias del hogar [Intille, S.S. 2002].

Un ejemplo de esta aproximación es una sala de estar que combina sensores que no invaden el espacio del usuario con proyectores ubicuos [Pinhaez, C. 2001] para crear un entorno donde la información se puede mostrar y manipular en casi cualquier superficie. De este modo es el propio entorno el que se convierte en el espacio donde se muestra la información y las aplicaciones compiten por utilizar este recurso. Una de estas aplicaciones es una herramienta que ayuda a los ocupantes del hogar a aprender un idioma extranjero. La herramienta no es disruptiva aunque está siempre presente, mostrando palabras en diversos espacios de la habitación. Cuando un usuario se interesa por una palabra sólo tiene que señalarla con un puntero láser. Esto hace que la herramienta pase a un plano más activo de interacción con el usuario [Intille, S.S. et al. 2003]. Así se consigue crear una interfaz persistente que no causa distracción auditiva o visual que pudiera desestabilizar la actividad que se está llevando a cabo en el entorno.

El interés de los proyectos que desarrollan se centra en detectar tres puntos temporales: el punto de decisión, el punto de comportamiento y el punto de consecuencia y en cómo utilizar sensores que detecten automáticamente estos momentos específicos para educar a la gente sobre cómo controlar su entorno. El objetivo se convierte en desarrollar algoritmos que reconozcan el momento adecuado y seleccionen la estrategia de presentación más idónea para el contexto dado, ayudándoles a tomar decisiones futuras, intentando evitar los perjuicios que se derivan de despojar a la gente de su sentido del control sobre los elementos [Rodin, J. and Langer, E. 1977].

Las actividades que se realizan en el entorno se detectan mediante sensores sencillos y de bajo coste instalados de forma ubicua, evitando otro tipo de sensores que pueden ser percibidos como agresivos, tales como cámaras o micrófonos (una aproximación totalmente opuesta a la de HAL). Las tasas de reconocimiento de las actividades varían sensiblemente, aunque pueden llegar a alcanzar cotas del 89% [Munguia Tapia, E. et al. 2004]

2.4.5 EasyLiving

Easyliving es el proyecto de Microsoft Research para desarrollar una arquitectura y las tecnologías necesarias en entornos inteligentes [Shafer, S.A.N. et al. 1998].

Los componentes del sistema EasyLiving incluyen un *middleware* que facilita la computación distribuida, un modelo del mundo que proporciona información de contexto basada en la localización de los elementos e individuos, percepción para recolectar información sobre el estado del mundo y descripción del servicio que permite la separación del control de dispositivos, la lógica interna y la interfaz de usuario. Esta separación permite cambiar de forma flexible los mecanismos de interacción sin que sea necesario modificar las aplicaciones subyacentes [Brumitt, B. et al. 2000].

El proyecto busca extrapolar las actividades que habitualmente se realizan en un escritorio al conjunto del entorno. Para ello consideran necesario que el sistema tenga un conocimiento más profundo del espacio físico desde la perspectiva sensorial y de control. Conocer esta información hace que la interacción con el usuario sea más natural. En su desarrollo se investigan las dificultades producidas al tener que percibir el entorno, como la necesidad de un modelo de incertidumbre, al tener que realizar análisis de datos en tiempo real o al tener que unir información, posiblemente contradictoria, de múltiples sensores.

Easyliving emplea una arquitectura de dispositivos aislados en comunicación directa, similar a la utilizada en otros proyectos como HAL o The Aware Home. Así emplean un *middleware* denominado InConcert que reduce el esfuerzo necesario para construir componentes individuales que se comuniquen en su entorno distribuido. InConcert proporciona envío de mensajes asíncrono, direccionamiento independiente de la máquina y un protocolo de mensajes basado en eXtensible Markup Language (XML).

Un elemento fundamental es el modelo geométrico que permite trabajar con múltiples tecnologías de percepción y que abstrae a las aplicaciones de los sensores específicos que se utilicen [Brumitt, B. and Shafer, S.A.N. 2001]. Este modelo se basa en entidades que representan la existencia de un objeto en el mundo físico y de medidas que se utilizan para definir relaciones geométricas entre entidades. Esta información proporciona un mecanismo para determinar los dispositivos que pueden ser utilizados para la interacción con el usuario y para ayudar en la selección de los dispositivos adecuados.

La información proporcionada al modelo geométrico se obtiene del espacio físico a través de diversos sensores, conectados a ordenadores que ejecutan componentes de percepción:

- Un componente de visión computacional se utiliza para localizar la posición de los individuos en el entorno, su identidad y su actividad [Krumm, J. et al. 2000]. Esta aproximación de alto nivel contrasta con los múltiples sensores de bajo nivel utilizados en el proyecto House_n.

- Un localizador de dispositivos inalámbricos móviles por radio frecuencia basado en la fuerza de la señal de puntos de acceso de la infraestructura conocidos.
- Un lector de huellas digitales que identifica a los usuarios del entorno. Este componente se utiliza en combinación con otros para asignar una identidad a los datos percibidos por los sensores.

Con esta tecnología se ha creado un entorno que proporciona al usuario acceso directo a los servicios disponibles. La disponibilidad de los servicios se determina mediante la intersección de la situación de los dispositivos actuales del usuario con la región de servicios. La interfaz de usuario se genera examinando cada descripción de servicio y mostrando los documentos XML adecuados. Otras aplicaciones implementadas son la posibilidad de establecer sesiones remotas con el entorno, utilizar ratones por radio frecuencia que interactúan con la pantalla más próxima o personalizar automáticamente las preferencias de audio y video cuando un usuario se autentifica en el entorno.

2.4.6 Interactive workspaces: iRoom

El proyecto Interactive workspaces, desarrollado por el Interactivity Lab, el Software Infrastructures Group y el Graphics Lab de la Stanford University, explora las nuevas posibilidades que se presentan en el empleo de la computación ubicua en el entorno del trabajo. Inicialmente se desarrolló con la idea de investigar la interacción persona-ordenador con pantallas de alta resolución. Actualmente emplean pantallas de gran formato, dispositivos multimodales o inalámbricos e integración de elementos móviles.

A partir de estas ideas construyeron su entorno de trabajo interactivo, denominado iRoom, así como una infraestructura software para este entorno y realizaron experimentos sobre interacción persona-ordenador en lugares de trabajo. Los principios básicos que guían este proyecto son [Johanson, B. et al. 2002]:

- Desde el primer momento se utilizó la iRoom como la sala principal de reuniones y se emplearon las herramientas software que se habían construido.
- Se eligió explorar nuevas formas de apoyo a las reuniones de grupo, tomando ventaja del espacio físico compartido.
- En lugar de tener una habitación que reacciona a los usuarios (como por ejemplo ocurre con HAL o EasyLiving) se eligió enfocarse en dejar a los usuarios ajustar el entorno según interaccionan con sus tareas.
- En lugar de investigar sistemas y aplicaciones exclusivamente en su espacio específico, decidieron investigar técnicas software que también se pudieran aplicar a entornos de trabajo configurados de forma diferente. Se pretendió así crear abstracciones estándares y metodologías de diseño de aplicaciones que se puedan emplear en cualquier entorno de trabajo interactivo.
- Tanto a nivel de interfaz como del desarrollo de software se intentó mantener una aproximación sencilla. Desde el punto de vista de interacción persona-ordenador se

intentó balancear la necesidad de trabajar con software y hardware diverso con la de proporcionar una interfaz de fácil manejo. Desde el punto de vista del software se procuró mantener las APIs (Application Programming Interfaces) tan sencillas como fuera posible, de modo que resultara fácil trasladar las bibliotecas a otros sistemas e incorporar a nuevos investigadores al proyecto.

Su entorno contiene pantallas de gran formato sensibles al tacto, cámaras, micrófonos, tecnología de red inalámbrica, diversos botones inalámbricos y otros dispositivos para interaccionar. Con estos elementos, el tipo de actividades que los usuarios pueden llevar a cabo en el entorno de trabajo interactivo son:

- Mover datos. Los usuarios en el entorno necesitan poder mover los datos entre las diferentes aplicaciones de visualización que se ejecutan en las pantallas de la habitación, entre los ordenadores portátiles o entre las PDAs (Personal Digital Assistants) que se encuentren en el entorno.
- Mover el control. Cualquier usuario debería poder controlar cualquier dispositivo o aplicación desde el lugar en el que se encuentre.
- Coordinación de aplicaciones dinámica. Las aplicaciones específicas que se necesitan para mostrar datos y analizar escenarios durante las sesiones de los equipos de trabajo son potencialmente diversas y deberían coordinarse con las demás de forma apropiada.

La interacción en el entorno de trabajo intenta minimizar la necesidad de crear nuevas interfaces y adaptarse a las características del usuario. Para ello se ha creado un mecanismo de distribución que simplifica añadir nuevas interfaces de usuario a un administrador centralizado de interfaces de usuario que serviría a múltiples entornos de trabajo diversos. Por otro lado un mecanismo de personalización permitiría a los usuarios trabajar con interfaces de usuario familiares incluso trabajando en entornos distintos [Ponnekanti, S.R. et al. 2002].

Los mecanismos de interacción con el entorno de trabajo se basan fundamentalmente en pantallas de gran formato permitiendo, por ejemplo, controlar menús y ejecutar comandos mediante lápices adaptados a los dispositivos [Guimbrètiere, F. and Winograd, T. 2000]. Dentro de esta interacción también se tiene en cuenta la interoperabilidad entre PDAs y las pantallas presentes en el entorno [Johanson, B. et al. 2000].

2.4.7 Otros proyectos

Los ejemplos vistos hasta ahora constituyen algunos de los proyectos más paradigmáticos, completos y referenciados que existen hasta el momento dentro del campo de los entornos inteligentes. Sin embargo, existen otros proyectos que también muestran diversos aspectos relacionados con esta categoría.

2.4.7.1 SmartOffice

SmartOffice es un proyecto desarrollado en el GRAVIR-IMAG Lab. del INRIA francés. Pretende crear una oficina inteligente que observa al usuario, para así anticiparse a sus intenciones y "enriquecer" su entorno para comunicar información de utilidad. El sistema se implica en las actividades de las personas para ayudarlas en sus tareas cotidianas. Para ello el entorno interacciona con los usuarios usando la voz, los gestos o los movimientos.

Este sistema [Le Gal, C. et al. 2001] está compuesto por módulos independientes que se integran en una única aplicación coherente. Para conseguir la integración de los módulos se utiliza un protocolo de integración flexible orientado a recursos por lo que los desarrolladores de cada uno de ellos no tienen que ocuparse de la comunicación a bajo nivel entre los módulos. Todos los módulos no necesitan tener conciencia sobre qué recursos pueden proporcionar los otros módulos. Se comunican con un supervisor que actúa como un servidor de recursos. El supervisor se programa usando un lenguaje basado en reglas en el cual la inclusión o la supresión de un módulo sólo requiere suprimir la regla correspondiente.

Se basa en las ideas de [Weiser, M. 1994], haciendo los ordenadores invisibles, sin exigir ningún tipo de adaptación por parte del usuario al no implicar nuevas formas de trabajar. Su arquitectura también sigue los pasos de HAL al proponer una arquitectura de control distribuida basada en la comunicación entre agentes. Su principal diferencia con el sistema de [Coen, M.H. 1997] es que un supervisor es responsable de la comunicación entre los agentes por lo que en lugar de proponer un modelo de comunicación orientado a los agentes, como ocurría en HAL, utiliza un modelo de comunicación orientado a recursos.

2.4.7.2 MavHome

El proyecto de casa inteligente MavHome (Managing an Adaptive Versatile Home) es un esfuerzo de investigación multidisciplinar de la University of Texas. Su objetivo es crear un hogar que actúe como un agente racional. El objetivo de este agente es maximizar el confort de sus habitantes y minimizar los costes operacionales. Para ello la casa debe ser capaz de predecir, razonar y adaptarse a sus ocupantes. El estado de la casa se percibe mediante sensores y se actúa en el entorno mediante controladores de los dispositivos.

La arquitectura de MavHome se basa en una jerarquía de agentes racionales que cooperan para conseguir los objetivos del hogar. Las tecnologías dentro de cada agente se separan en cuatro capas [Cook, D.J. et al. 2003]:

- La capa de decisión selecciona acciones para que sean ejecutadas por el agente basándose en la información suministrada por las otras capas a través de la capa de información.
- La capa de información recoge, almacena y genera conocimiento útil para tomar decisiones.

- La capa de comunicación facilita la comunicación de información, solicitud y consultas entre los agentes.
- La capa física contiene el hardware del hogar, incluyendo los dispositivos, sensores y la red de comunicación.

En el proceso de percepción los sensores monitorizan el entorno y, si resulta necesario, transmiten información a otro agente a través de la capa de comunicación. Una base de datos registra la información a través de la capa de información, actualiza su listado de conceptos aprendidos y de predicciones y alerta a la capa de decisión de la presencia de nuevos datos.

En la ejecución la capa de decisión selecciona una acción y transmite la decisión a la capa de información. Después de actualizar la base de datos la capa de comunicación envía la acción al controlador apropiado para su ejecución. Si el controlador es otro agente, éste recibe el comando a través del control como información percibida y debe decidir sobre el mejor método de ejecución de la acción deseada.

Las características de MavHome incluyen recopilación de actividades en una base de datos, predicción de las actividades de sus habitantes, identificación de los habitantes mediante la observación de las actividades, predicción de la movilidad y control inteligente del hogar. Con estas habilidades, el hogar puede controlar muchos aspectos del entorno, como el clima, la iluminación, el mantenimiento y el entretenimiento.

Según su aproximación, la predicción de las siguientes actividades que se van a llevar a cabo y su automatización puede reducir la cantidad de interacciones requeridas por sus ocupantes y reducir el consumo de energía. Estas mismas habilidades también se pueden utilizar para permitir la detección de comportamientos poco habituales en sistemas de control de la salud en el hogar (una aproximación similar mostraba The Adaptive House) y para seguridad en el hogar [Cook, D.J. and Youngblood, M. 2004].

2.4.7.3 The KidsRoom

The KidsRoom es un proyecto desarrollado en el Media Lab del MIT que pretende construir un entorno narrativo interactivo [Bobick, A. et al. 1997] que pueda ser utilizado por niños. El sistema responde a las acciones de los usuarios en un espacio físico real "enriqueciendo" el entorno con gráficos, vídeo, efectos de sonido, luz, música y narraciones. Los modos de entrada de información son cámaras y micrófonos no obstructivos, por lo que los usuarios no tienen que portar sensores, displays, ropas especiales o micrófonos. Se basa en la idea de que una sola cámara puede realizar las funciones de múltiples sensores distribuidos (coincidiendo con el punto de vista de HAL.

El sistema sumerge a los niños en un entorno narrativo en el que varios niños a la vez pueden interaccionar mediante la voz y los gestos mientras las cuatro paredes de la habitación actúan como *displays* gigantes. El proceso narrativo se ha diseñado de forma cuidadosa para no crear expectativas que no se puedan satisfacer. El sistema no sólo

mide la posición de los usuarios sino que reconoce sus acciones utilizando también el contexto.

Los objetivos buscados en la construcción de este sistema son [Bobick, A. et al. 1997]:

- Mantener la acción en un espacio físico.
- Usar percepción remota basada en la visión para estimular la interacción natural no intrusiva.
- Construir un sistema que funcione de forma efectiva con múltiples usuarios.
- Reducir la fragilidad de la percepción, permitiendo al sistema usar y manipular el contexto.
- Crear un espacio que sea inmersivo, atractivo y absolutamente automático y con el que se pueda interaccionar de forma natural sin necesidad de instrucciones previas.

2.4.7.4 The Reactive Room

The Reactive Room es un proyecto que se llevó a cabo en la Universidad de Toronto para desarrollar una sala de conferencias basada en las ideas de computación ubicua y transparencia que reaccione según las intenciones del usuario. La habitación está equipada con cámaras, monitores, micrófonos y altavoces de modo que permita a usuarios remotos participar en reuniones. Su construcción se encuentra motivada por las siguientes ideas [Cooperstock, J.R. et al. 1995]:

- Intentar reducir la complejidad del funcionamiento de la habitación. En especial, reducir la cantidad de conocimiento explícito requerido por el usuario para utilizarla de forma efectiva.
- Reducir la intromisión en las reuniones de la gestión de aspectos operacionales de la habitación.
- Evitar que las acciones de la habitación se manejen por un operador.

El operador que conduce la habitación es la propia tecnología de la habitación, en lugar de un operador humano. El sistema recopila información sobre el contexto para poder deducir las acciones de los usuarios basándose en su comportamiento. Para conseguirlo, se analizan las salidas de los sensores para determinar cuándo se deberían llevar a cabo ciertas acciones por el entorno, esto es, cómo debería reaccionar el entorno. Tal y como ocurre en The Adaptive House, los usuarios tienen preferencia sobre las acciones llevadas a cabo por el sistema. Además las reglas de interacción resultan naturales a los usuarios y no requieren comprensión de la tecnología por parte de estos. Esto implica que también exista un mecanismo de diagnósticos que proporcione información sobre un problema del sistema.

2.4.7.5 Classroom 2000

Classroom 2000, actualmente denominado eClass, es un proyecto desarrollado en el Graphics, Visualization and Usability Center del Georgia Institute of Technology por algunos de los miembros a cargo de Domisilica y The Aware Home. Si bien estos y los otros sistemas se basaban en el entorno del hogar o la oficina, el interés de Classroom 2000 es el de desarrollar entornos computacionales para mejorar la actividad dentro de un aula [Abowd, G.D. et al. 1996]. El proyecto investiga el impacto de la computación ubicua en la educación universitaria.

En un aula tradicional, el profesor escribe en una pizarra y utiliza diapositivas para impartir su clase. El problema con este sistema es que el estudiante se ve forzado a copiar todo lo que el profesor escribe en su propio cuaderno si no quiere que esa información se pierda. En Classroom 2000, el sistema proporciona al profesor y a los alumnos tecnología basada en escritura, servicios de audio o acceso a la Web para apoyar la captación de la actividad que se lleva a cabo dentro de una clase. Esta actividad obtenida mediante computación ubicua se integra para que pueda ser posteriormente revisada y mejorada mediante las interacciones del grupo [Abowd, G.D. et al. 1998].

El objetivo se consigue proporcionando al aula una pizarra electrónica que permite al profesor impartir una clase normal, utilizando además una presentación o una serie de páginas Web como fondo. El sistema capta muchas de las actividades del profesor asignándoles momentos temporales. Además, la habitación está equipada con una infraestructura de grabación digital. Como resultado, se obtienen unos apuntes accesibles mediante la Web que coordinan el audio y el vídeo obtenidos del profesor y las URLs (Uniform Resource Locators) a las que accedió el profesor durante la clase [Brotherton, J.A. and Abowd, G.D. 1998].

El sistema pretende aliviar a los alumnos de la carga de obtener apuntes mientras se realiza la clase para que así puedan involucrarse más directamente en el transcurso de la misma.

2.5 Conclusiones obtenidas sobre entornos inteligentes

A partir del estudio de los sistemas de computación ubicua y los entornos inteligentes mostrados en las secciones anteriores se puede realizar una clasificación de los entornos inteligentes según cuatro parámetros principales: espacio de aplicación, modelo de representación, mecanismos de obtención de la información e interacción con los usuarios.

Según el espacio de aplicación los entornos inteligentes se pueden dividir en:

- Entornos móviles. Estos sistemas basan su funcionalidad en el empleo de elementos móviles que pueden ser transportados por los usuarios (como teléfonos móviles o PDAs, o el empleo de ordenadores que se llevan puestos junto con la ropa, *wearable computers*). En esta aproximación, la interconexión y el intercambio de información

entre los diferentes elementos que componen el entorno resulta una característica fundamental. El sistema debe ser capaz de descubrir de forma dinámica qué elementos entran o dejan de formar parte del entorno, además de proporcionar un mecanismo de descubrimiento automático de nuevos elementos y de intercambio estándar de información entre los mismos.

- Espacios embebidos. Se trata de entornos que rodean al usuario y en donde los elementos que lo componen resultan, en gran medida, invisibles al mismo. Estos espacios no resultan tan dinámicos como los entornos móviles aunque deben ser capaces de adaptarse automáticamente a la configuración que tengan en cada momento. Además, deben garantizar el empleo de aplicaciones y dispositivos heterogéneos, sin que ellos repercuta en la funcionalidad del sistema.

En ambos casos se ha de permitir que las aplicaciones y dispositivos aparezcan y desaparezcan o se añadan y eliminen de forma fácil y automática. Un requisito necesario para dotar al entorno de esta cualidad consiste en establecer una ontología del mundo, donde se represente el estado del mismo y que además sirva como modelo para estandarizar las descripciones de los recursos. A su vez, se debería definir una interfaz entre las aplicaciones y los dispositivos del sistema, permitiendo modificar los recursos *software* y *hardware* sin variar la funcionalidad del sistema. Este grado de independencia entre aplicaciones y dispositivos hace necesario que la información se intercambie entre ellos de forma homogénea. Todos estos requisitos son consecuencia de una de las características principales de los entornos inteligentes y que es común a todos ellos: su dinamismo. El entorno debe poder adaptarse fácilmente a los múltiples cambios que en él se pueden producir.

Para la representación del entorno y comunicación entre sus componentes se utiliza una capa *middleware*. Sin embargo, existen puntos de vista divergentes sobre qué aproximación es más adecuada para la implementación de este *middleware*. Basándose en el modelo de representación empleado los entornos se pueden clasificar en:

- Distribuidos. En estos sistemas la solución propuesta para representar e intercambiar la información del entorno consiste en un modelo de agentes distribuidos que interaccionan entre sí. Cada agente es independiente y se ocupa de realizar una tarea concreta e independiente. Esta es por ejemplo la aproximación que se emplea dentro del proyecto HAL (ver apartado 2.4.1). La complejidad de estos sistemas se encuentra en posibilitar la comunicación entre los diferentes agentes y su colaboración para realizar tareas avanzadas.

- Centralizados. Es estos entornos la capa *middleware* se encuentra centralizada en un repositorio común donde se depositan todos los datos que puedan resultar de interés. Un ejemplo de esta aproximación es The Aware Home (ver apartado 2.4.2).Como aspecto positivo, al centralizar toda la información, la comunicación entre aplicaciones y dispositivos se realiza de forma unificada y estándar. Se simplifica considerablemente el acceso a los recursos y servicios del entorno y se elimina la dificultad de presentar y descubrir un nuevo elemento al resto del entorno. Como

factor negativo, si no se gestiona correctamente, puede suponer un cuello de botella en el sistema o agotar los recursos existentes en un único componente centralizado.

Una vez que se ha conseguido especificar una correcta representación del entorno y se ha provisto la comunicación entre aplicaciones y dispositivos resulta necesario dotarle de mecanismos de obtención, representación y manejo del contexto (esta idea resulta fundamental en todos los proyectos vistos previamente, aunque se podrían destacar House_n, ver apartado 2.4.4, o The Adaptive House, ver apartado 2.4.3). La forma más común para obtener la información de contexto es la de recopilarla mediante el uso de sensores. De acuerdo con los mecanismos de obtención de la información, los entornos inteligentes se pueden dividir entre los que utilizan:

- Sensores de alto nivel. Son aquellos que emplean cámaras y micrófonos (en muchas ocasiones en conjunción) para obtener información contextual del entorno. Este es el caso de, por ejemplo, HAL. Se basan en la idea de que uno sólo de estos sensores es capaz de procesar la misma información que múltiples sensores de bajo nivel. Sin embargo, sus principales problemas vienen de la dificultad de interpretar la información percibida y por el mayor coste que éstos presentan. Además, cabe destacar el rechazo que puede suponer para muchos individuos el uso de cámaras y micrófonos dentro de su ámbito de convivencia.

- Sensores de bajo nivel. Se trata de los sistemas que utilizan pequeños sensores (por ejemplo de presencia, presión, etc.), normalmente embebidos dentro del entorno e imperceptibles para el usuario. El uso aislado de este tipo de sensores no suele proveer una información de contexto precisa y de gran interés pero la conjunción numerosos sensores (puede que incluso cientos) puede proporcionar una información completa y fiable. Como ejemplo de esta aproximación se encuentra el proyecto House_n. A su favor tienen la facilidad de instalación, mantenimiento unitario y manejo, además de su poca intromisión dentro del entorno. En su contra se halla la necesidad de utilizar múltiples sensores. Esto implica posibles dificultades de comunicación, mantenimiento de un entorno con múltiples dispositivos e interpretación de posibles informaciones contradictorias.

A pesar de esta división, ambas aproximaciones no tienen que ser necesariamente excluyentes, por lo que el empleo de diferentes mecanismos puede resultar de utilidad para resolver problemas distintos.

A partir de la información obtenida de los sensores los entornos ofrecen dos posibles enfoques que se basan en el grado de interacción que proporcionan a los usuarios:

- Proactivos. Son aquellos en los que la finalidad principal del sistema se encuentra en determinar qué acciones está realizando el usuario y cuáles son sus necesidades para responder a ellas de forma anticipada. La interacción con este se limita a la respuesta que el usuario proporciona al sistema tras una decisión. Esta respuesta suele servir de refuerzo para la toma de futuras decisiones. Generalmente, estos entornos descartan mecanismos de interacción a más alto nivel. Así ocurre por ejemplo con The Adaptive House.

- Interactivos. Se trata de sistemas donde se prima la posibilidad de establecer una comunicación natural e intuitiva entre el entorno y sus ocupantes. Se pretende que la interacción sea tan parecida que sea posible a la que realizan dos humanos entre sí. Para ello se utilizan técnicas de reconocimiento de escritura, de interacción mediante diálogos orales o de reconocimiento de gestos. Tal es el caso de HAL.

Independientemente de la aproximación que se elija, un aspecto fundamental para procesar correctamente la información y proporcionar una respuesta adecuada es el uso del contexto. Esta característica es común tanto en los sistemas proactivos como en los interactivos ya que permite adaptar el comportamiento del entorno a todas las posibles situaciones que se pueden dar en el mismo. Por lo tanto la capacidad de adaptación del sistema a sus ocupantes y a su situación actual resulta fundamental dentro de cualquier entorno inteligente.

Otra característica común a los entornos inteligentes desarrollados hasta el momento ha sido la de implementar el mismo en un laboratorio que sirva como infraestructura científica desde las primeras fases de desarrollo (especial énfasis en este aspecto realiza el proyecto iRoom, ver apartado 2.4.6). Dado que los entornos inteligentes son un nuevo espacio de convivencia de las personas resulta fundamental comprobar su aceptación, la adecuación de las ideas a los modos de uso de sus ocupantes y la adaptación de estos a un entorno real dinámico y heterogéneo.

2.6 El sonido como interfaz de usuario

Como se ha visto en el apartado anterior, el sonido puede ser un elemento fundamental dentro de los entornos inteligentes. Para la construcción de un entorno inteligente en el que se utilice el sonido como interfaz de usuario podrá ser necesaria la presencia de:

Micrófonos, que se pueden utilizar para:

- Identificar los sonidos percibidos por el entorno (pisadas, puertas, etc.).
- Reconocer la voz y obtener una representación en forma de texto de las palabras pronunciadas por los habitantes del entorno.
- Identificar al hablante por medio de la voz.
- Detectar la posición del usuario en el entorno.

Gestores del sonido percibido y de diálogos, que podrían incluir:

- Gestores del flujo del sonido que establezcan una concordancia contextual entre los sonidos percibidos y las acciones que se llevan a cabo.
- Analizadores, *parsers* y gramáticas que identifiquen el texto reconocido dentro de un diálogo.
- Gestores del flujo de diálogo que hagan posible establecer una conversación coherente con el entorno.

- Traductores del idioma que posibiliten hacer traducciones dentro de un entorno multilenguaje.

Altavoces para comunicarse con el usuario del entorno, lo que podría implicar:

- Generadores de sonido que reproduzcan sonidos en un entorno de realidad "enriquecida" (augmented reality).
- Sintetizadores de voz que permitan generar una voz reconocible por el usuario a partir del texto que se quiere transmitir.

2.7 Entrada de sonido

Evidentemente los micrófonos constituyen un elemento fundamental de los entornos inteligentes en los que se interactúa mediante la voz. Los micrófonos se ocupan de capturar el sonido que posteriormente será procesado por otros componentes dentro del sistema.

2.7.1 Captura del sonido

La utilización de micrófonos para capturar el sonido se puede realizar de forma obstructiva para el usuario o de manera completamente no obstructiva:

- En el primero de los casos, el usuario debe portar el micrófono consigo. Esta práctica redunda en un considerable aumento en la calidad de la captura del sonido. Un simple micrófono con cancelación de ruido es suficiente para obtener resultados altamente satisfactorios en la captura. Sin embargo, obliga al usuario a llevar consigo un dispositivo que no forma parte de su indumentaria habitual, esto es, las personas tienen que portar un dispositivo específico para interaccionar con el entorno.
- Por otro lado, si se desea poder interactuar de forma no obstructiva se debe utilizar un micrófono de ambiente. El problema que conlleva este tipo de micrófonos radica en una menor calidad en la captura del sonido percibido. En contrapartida se obtiene un entorno más confortable para el usuario, proporcionando una interacción transparente con el mismo.

El problema de la calidad de percepción de los micrófonos de ambiente se puede solventar parcialmente utilizando más de un micrófono para reconocer el sonido. El uso de varios micrófonos dentro de un mismo entorno para capturar un único sonido reduce considerablemente la posibilidad de error en el posterior reconocimiento del mismo [Lleida, E. et al. 1998].

Además, un conjunto de micrófonos resulta especialmente útil cuando se trabaja con varias fuentes de sonido o cuando éstas no son fijas. Este conjunto de micrófonos permite seguir a una fuente de sonido móvil o cambiar en la percepción de distintas fuentes de sonido que se emitan de forma alternativa o simultánea [Brandstein, M.S. et al. 1996].

Una aplicación clara que presenta el uso de diferentes micrófonos es la localización del sonido. El uso de un mínimo de cuatro micrófonos [Guentchev, K.Y. and Weng, J.J. 1998] hace posible situar fuentes de sonido arbitrariamente colocadas en un entorno.

2.7.2 Reconocimiento del habla

A pesar de las posibles utilidades descritas en el apartado anterior, el uso más común de un micrófono es el de la captura de la voz. Esto permite realizar posteriormente reconocimiento del habla. El reconocimiento del habla consiste en obtener una representación en texto de las palabras pronunciadas por el usuario.

2.7.2.1 Sistemas de reconocimiento del habla

La clasificación más básica que se puede realizar entre los sistemas de reconocimiento del habla es [Lai, J. and Vergo, J. 1997]:

- Sistemas de reconocimiento discreto. En estos sistemas es necesario que el usuario inserte una pausa entre palabra y palabra. De este modo, resulta mucho más sencillo para el sistema reconocer las palabras.
- Sistemas de reconocimiento continuo. En los que el usuario puede hablar de forma completamente natural sin tener que preocuparse de introducir pausas.

Existen otras divisiones para clasificar estos sistemas. Se puede distinguir entre sistemas que necesitan un entrenamiento previo para adaptarse a la voz del usuario y sistemas independientes del hablante; o sistemas que sólo son capaces de reconocer comandos y frases fijas y sistemas que hacen uso de gramáticas [Yankelovich, N. and Lai, J. 1998].

Existen diversas herramientas que permiten reconocer las frases pronunciadas por los usuarios. De entre ellas, varias son comerciales y consiguen una alta calidad de reconocimiento utilizando una combinación de sistemas discretos y continuos [Benet, B. 1995]. Para refinar el proceso de reconocimiento estas herramientas vienen dotadas de un vocabulario por defecto, un modelo de lenguaje y un conjunto de pronunciaciones; de modo que sólo son capaces de reconocer las palabras pertenecientes a su vocabulario. Además cuanto más restringido sea el dominio, más fácil resultará para la herramienta poder reconocer lo pronunciado por el usuario.

2.7.2.2 Problemas en el reconocimiento del habla

Las herramientas de reconocimiento del habla producen una serie de errores que se pueden dividir en tres categorías [Schmandt, C. 1994]:

- Errores de rechazo. Se producen cuando el reconocedor es incapaz de determinar qué ha dicho el usuario.
- Errores de sustitución. Aquellos en los que el resultado proporcionado por el reconocedor difiere de la entrada que éste ha recibido.

- Errores de inserción. En los que el reconocedor interpreta el ruido como una oración válida.

Para intentar restringir lo máximo posible estos errores se utilizan diferentes métodos con resultados satisfactorios:

- Mediante un entrenamiento previo por parte del usuario se puede alcanzar un rendimiento aceptable [Karat, C. et al. 1999].
- Del mismo modo que el contexto ayuda a la comprensión del habla por parte de los humanos, se puede utilizar éste para obtener mejores resultados del reconocedor [Nagao, K. and Rekimoto, J. 1995]. Por ejemplo, se puede cambiar la gramática que se emplea en cada momento dependiendo del contexto [Coen, M.H. et al. 1999b, Yan, H. and Selker, T. 2000]. De este modo se consigue restringir al máximo el dominio, por lo que resulta mucho más sencillo predecir una palabra dada la palabra que la precede [Lai, J. and Vergo, J. 1997]. Esta predicción de palabras (tanto habladas como escritas [Garay-Vitoria, N. and González-Abascal, J. 1997]) no sólo se utiliza para conseguir una mejora en la tasa de reconocimiento, sino que presenta una considerable ayuda para personas con discapacidades [Alm, N. et al. 1992].
- Hay que realizar una correcta configuración del micrófono, ajustando por ejemplo su ganancia, y se deberá tener en cuenta el correcto uso del mismo por parte del usuario (en entornos donde el micrófono actúa de forma obstructiva) o su colocación adecuada dentro del entorno (en aquellos que sean no obstructivos).
- Se pueden utilizar diferentes micrófonos para comparar los resultados obtenidos por el reconocedor en cada uno de ellos y reducir de esta manera los posibles errores.
- Estos sistemas se ven seriamente limitados en su capacidad de reconocimiento cuando se trabaja en entornos ruidosos. La tasa de reconocimiento para un sistema independiente del hablante en un entorno silencioso, que puede ser del 98% para vocabularios pequeños, se puede reducir hasta solamente un 40% en entornos ruidosos [Gómez, P. et al. 1998]. De este modo, se hace necesario un tratamiento especial de las señales si se va a trabajar en un entorno en el que la aparición de ruido puede ser un inconveniente para una correcta percepción por parte del reconocedor.

2.8 Gestión del sonido

El sonido, como interfaz de usuario, puede suponer una valiosa alternativa a la información visual al ser capaz de proporcionar información contextual con libertad de movimientos. Además el sonido permite oír aquello que no podemos ver y evita la necesidad de que el usuario tenga que cambiar su foco visual cada vez que se produce un nuevo suceso.

El sonido, como elemento dentro de una interfaz de usuario, se puede utilizar fundamentalmente de dos formas diferentes:

- Como señales de audio que proporcionen al usuario conciencia directa o indirecta sobre qué ocurre en un entorno determinado.
- Convirtiendo a la interfaz en una interfaz hablada con la que se interactúa mediante comandos fijos o mediante diálogos y se obtiene una respuesta audible.

Ambos usos del sonido cuentan con apoyo dentro del campo de la investigación [Arons, B. and Mynatt, E. 1994] para explotar sus ventajas y conseguir hacer las interfaces más accesibles, intuitivas y cómodas para el usuario. Estas aproximaciones no son excluyentes sino que se pueden combinar dentro de un mismo sistema.

2.8.1 Señales de audio

El mundo físico se puede "enriquecer" con señales de audio que permitan una interacción pasiva con el usuario [Mynatt, E.D. et al. 1997]. Estas señales pueden resumir información sobre la actividad de un colega, notificar el estado del e-mail, el comienzo de una reunión o recordar al usuario tareas que debe realizar. Las señales de audio de esta manera se usan para generar conciencia del contexto del usuario. La información se puede enviar a un usuario móvil que se encuentre en cualquier lugar del entorno físico en cualquier momento sin necesidad de acceder a un dispositivo concreto para conocerla. Con este método se consigue crear un modo de interacción no intrusivo y que no requiere la participación activa del usuario.

Siguiendo estas ideas se puede utilizar el sonido para transmitir información compleja basándose en cómo la gente escucha los sucesos cotidianos. Se pueden utilizar los atributos de los sucesos cotidianos para crear atributos de sucesos computacionales. Aparecen así los *iconos de audio* [Gaver, W.W. 1991]. Los *iconos de audio* son sonidos de ambiente diseñados para expresar información análoga a la que producen los sonidos cotidianos. Los *iconos de audio* presentan varias cualidades interesantes como método para proporcionar información sobre sucesos:

- El sonido como medio es una forma valiosa para proporcionar información que proviene de diferentes posiciones.
- El audio no hablado a menudo distrae en menor medida y, para ciertas situaciones, más eficaz que el habla.
- Los sonidos cotidianos pueden llegar a adaptarse a los sucesos que intentan representar de forma más fidedigna que otro tipo de señales sonoras.
- Se pueden diseñar los *iconos de audio* para que presenten información de un modo casi subliminal.

Las señales de audio que permiten el movimiento de una persona en un entorno físico proporcionando información de forma directa o subjetiva al usuario son estudiadas por numerosos investigadores [Mynatt, E.D. et al. 1997] y artistas. Estas señales se utilizan actualmente, por ejemplo, en recorridos de museos y galerías. También se pueden

utilizar combinadas con otros medios para crear conciencia del entorno en el que se encuentran.

2.8.2 Retos en el desarrollo de interfaces del habla

Desde otro punto de vista, el uso de los micrófonos y, fundamentalmente, el avance conseguido en las herramientas de reconocimiento de voz ha provocado un interés cada vez mayor en el desarrollo de *interfaces de usuario orales* (SUI). Este tipo de aplicaciones fueron concebidas como sustituto de las *Interfaces gráficas de usuario* (GUI), pero poco a poco se fueron adaptando a situaciones específicas donde ofrecían ventajas concretas sobre los entornos visuales. Estas interfaces son de gran utilidad en entornos telefónicos [Wolf, C.G. and Zadrozny, W. 1998], sistemas de navegación por Internet, sistemas en los que las manos y los ojos están ocupados (por ejemplo asistentes para la conducción [Geutner, P. et al. 1998]) o para la creación de habitaciones y entornos con interacción en un modo natural.

La creación de estos sistemas con interfaces de habla, requiere afrontar nuevos problemas y situaciones. Los retos que presenta este nuevo tipo de desarrollo se describen extensamente en [Yankelovich, N. et al. 1995].

2.8.2.1 Conseguir una correcta conversión de sistemas anteriores en SUIs

El vocabulario usado en sistemas con interfaces de usuario tradicionales (por ejemplo GUIs) no se puede transferir directamente a las SUIs. Para estos nuevos casos resulta mucho más conveniente utilizar frases que cumplan las convenciones establecidas dentro de las conversaciones. Sin embargo, para conseguir una alta capacidad de reconocimiento el sistema debe intentar conducir al usuario a respuestas que se encuentren dentro de sus capacidades [Hansen, B. et al. 1996].

El modo de organización y presentación de la información puede presentar complicaciones dentro de un entorno conversacional. Si la SUI no cuenta además con un apoyo visual (como en el caso de aplicaciones telefónicas) esta complicación puede resultar especialmente difícil de manejar.

El flujo de información en una SUI puede resultar confuso para el usuario. Si el diálogo implica subdiálogos puede ser difícil determinar en qué punto se encuentra la conversación, si ya se ha retornado a un estado anterior o si las respuestas corresponden al diálogo en que se suponía estar.

2.8.2.2 Simular por parte del sistema una conversación de la forma más real y natural que sea posible

Uno de los principales desafíos de este tipo de aplicaciones consiste en poder simular el papel de hablante o de oyente de forma lo suficientemente convincente como para producir una comunicación satisfactoria con el usuario.

Un elemento importante en toda conversación es la entonación. Actualmente se está trabajando en conseguir herramientas de síntesis con unos niveles de entonación

similares a los humanos, en los que la voz no suene metálica y sea lo más natural posible.

Las pausas que se realizan en las conversaciones constituyen otro de los factores que se deben tener en cuenta dentro de los diálogos. Se ha de procurar que las pausas generadas por los retrasos en el proceso de reconocimiento sean lo suficientemente cortas para que sean percibidas como naturales.

Por último, no todos los sistemas permiten interrumpir al sintetizador mientras está pronunciando una oración, lo que puede repercutir en una falta de naturalidad en el diálogo. El sintetizador también debería poder aumentar o disminuir la velocidad de su discurso para simular, en mayor medida, el habla humana [Marx, M. and Schmandt C. 1996].

2.8.2.3 Gestionar de forma adecuada los errores que se pueden provocar durante el reconocimiento del habla

La mayor parte de los errores que se originan en los sistemas que utilizan SUIs se producen durante el reconocimiento del habla. Para conseguir que la interacción resulte lo más natural y adecuada posible se han de manejar correctamente los errores producidos por los reconocedores del habla. Esto implica que además de los métodos expuestos en el apartado 2.7.2.2, se deben tener en cuenta nuevas técnicas dentro de la conversación [Wolf, C.G. and Zadrozny, W. 1998] dependiendo del error producido:

- Los errores de rechazo (cuando el reconocedor no puede determinar qué se ha pronunciado) se pueden gestionar de dos maneras posibles:
  - De forma conjunta, sin hacer distinciones entre errores, o
  - Suponiendo la posible naturaleza de los errores de manera que se pueda proporcionar una asistencia adaptada y progresiva. Este método permite reconducir al usuario en la conversación de modo que repita la frase con otras palabras o intente hablar de forma más clara para el reconocedor.
- Los errores de sustitución (en los que se reconoce una expresión distinta a la pronunciada) pueden producir fallos inesperados en el comportamiento del sistema. Este tipo de errores hace que pueda resultar necesario asegurarse de que se ha comprendido correctamente lo dicho por el usuario. Como la continua comprobación de este tipo de errores supone una considerable carga dentro de la conversación lo mejor es que se realice solamente en momentos críticos o en los casos en los que se tenga alguna sospecha de que la entrada no corresponde con lo esperado.
- Los errores de inserción (cuando el ruido produce una frase) se pueden tratar con algunos de los métodos anteriores, o incluso desconectando el micrófono o el reconocedor en determinados momentos de la conversación. Muchos sistemas optan por una interfaz *click-to-speak* (en los que el micrófono o el reconocedor no están operativos hasta que el usuario los activa explícitamente) en contraposición con una

interfaz de estilo *open-microphone* (donde el micrófono y el reconocedor funcionan continuamente). Con el método *click-to-speak* se consigue descartar hasta un 12,4 % de palabras ininteligibles en un sistema *open-microphone* [Oviatt, S. et al. 1997] aunque se restringe considerablemente la libertad de interacción con el sistema.

Además de los métodos expuestos en los tres puntos anteriores, según el planteamiento de [Marx M. and Schmandt, C. 1996] se consigue una mejor gestión de los errores si éstos se tratan de forma individualizada y si se mantiene un registro con los últimos errores ocurridos. Este método permite tratar a los nuevos errores de forma óptima dependiendo de qué clase de error se trate y cuáles hayan sido los resultados sobre acciones anteriores.

2.8.2.4 Ajustarse a los nuevos problemas generados debido a la naturaleza del habla

La ausencia de apoyo visual en muchos de estos sistemas también puede repercutir negativamente en la capacidad de control por parte del usuario y en la cantidad de información que le puede ser transmitida. Por ejemplo, las pausas producidas por el sistema durante el procesamiento de la información pueden no ser bien entendidas por los usuarios.

Dado que el habla es un método no persistente de transmisión de la información (lo que puede provocar una carencia de retención por parte del usuario) las oraciones que transmite el sistema han de ser breves, concisas e informativas. Se ha de evitar utilizar palabras que no sean necesarias o que resulten extrañas o repetitivas. Aplicando un principio de [Grice, H. 1975], el sistema debe proporcionar la menor cantidad de información posible, aunque nunca menos de la necesaria.

Los silencios que produzca el sistema pueden generar situaciones de inestabilidad para el usuario. Los silencios se han de evitar siempre que su presencia no corresponda a la situación esperada dentro de una conversación.

2.8.3 Adaptación del usuario a nuevas interfaces habladas

Un elemento fundamental para el desarrollo de estos sistemas es que resulten naturales e intuitivos para aquellos que los utilizan. Estas nuevas interfaces se han de implementar de modo que al usuario le suponga una mínima carga de memorización y puedan ser accesibles como si la interacción se realizara con un humano. Sin embargo, tal y como se discute en [Yankelovic, N. 1996], las propias limitaciones tecnológicas imponen ciertas restricciones:

- Como ya se ha comentado en el apartado 2.7.2, el reconocedor del habla sólo permitirá reconocer aquellas palabras que pertenezcan a su diccionario.
- Para evitar que se produzcan confusiones (demasiadas palabras dentro de un mismo diccionario con la misma posibilidad de ser reconocidas) las gramáticas se han de restringir para obtener unas tasas de reconocimiento aceptables.

Esta situación propicia que el usuario diga cosas que el sistema no es capaz de entender. Además, restringir en exceso la clase de oraciones que se pueden pronunciar para obtener un determinado servicio o respuesta puede implicar que los usuarios no sepan cómo han de comunicarse con el sistema para establecer un diálogo.

Las expectativas que la gente puede tener de los sistemas conversacionales son muy amplias. Se pueden producir dos situaciones diferentes:

- Que los usuarios consideren que el sistema es capaz de comportarse con un nivel de entendimiento igual al que tendría un humano.
- Que los usuarios esperen un comportamiento limitado del mismo, confiando en que éste les guíe sobre las diferentes formas en las que pueden interaccionar.

Se pueden tomar diferentes enfoques para guiar a los usuarios en cómo pueden empezar a interactuar con el sistema. Desde la estrategia de [Yankelovich, N. et al. 1995] en la que los usuarios reciben un listado de opciones y comandos de ejemplo, hasta la de [Marx, M. and Schmandt C. 1996] en la que, de forma no intrusiva, se utilizan los propios recursos lingüísticos de los que dispone el sistema para guiar al usuario en el aprendizaje sobre cómo desarrollar las conversaciones.

La última motivación de muchos de estos sistemas es hacer que los servicios computacionales sean accesibles no sólo para expertos en tecnología sino también para ciudadanos sin conocimientos previos y sin necesidad de un entrenamiento especial [Tran-Huy, K. and MELISSA Consortium 1998]. Desde este punto de vista, cualquier usuario (expertos, público en general, discapacitados, personas mayores, etc.) podrá interactuar en su idioma natal con diferentes aplicaciones informáticas, lo que puede suponer una gran ventaja en el desarrollo de sus tareas cotidianas.

2.9 Gestión de diálogos

Uno de los aspectos fundamentales en el desarrollo de interfaces del habla es la gestión de diálogos. Por gestión de diálogos se entiende el manejo y comprensión de lo dicho por el usuario, de modo que el sistema pueda proporcionar una respuesta correcta y distinta para cada caso. El sistema ha de ser consciente del contexto en el que se encuentra y utilizarlo de forma inteligente para procesar la entrada recibida y producir una respuesta precisa. Esta respuesta podrá ser distinta para una misma entrada si el contexto y el entorno difieren. De este modo, el usuario y el sistema pueden entablar una conversación coherente con diferentes niveles de profundidad dentro de un entorno más o menos restringido.

2.9.1 Primeros hitos en la gestión de diálogos

Dentro del procesamiento de lenguaje natural, el área de investigación encargada de las conversaciones con el ordenador comenzó su desarrollo en los inicios de los años setenta. Si bien durante esa etapa empezaron a prosperar proyectos que no correspondían exactamente con sistemas de gestión de diálogos [Lehnert, W.G. 1977],

muchos de ellos sentaron las bases de la posterior evolución de los sistemas que permiten entablar conversaciones con distintos niveles de restricción.

Fue durante esos años cuando apareció PARRY [Parkinson, R.C. 1977], uno de los primeros intentos en modelación del conocimiento. El sistema simula los procesos mentales de un paranoico. Ideado por Colby en Stanford, produce un comportamiento que fue clasificado como paranoia por varios psicoanalistas que conversaron con él. Sin embargo, este proyecto ha quedado siempre eclipsado por la fama que llegó a alcanzar un sistema antecesor, ELIZA [Weizenbaum, J. 1966], capaz de mantener una conversación sin apenas "entender" las frases que utiliza el usuario y generando respuestas mediante la combinación de frases almacenadas y la transformación de las frases recibidas. Para aislar el dominio de la aplicación se utiliza un *script*. Así, mediante el uso de diferentes *scripts* se ha conseguido adaptar satisfactoriamente una versión mejorada del sistema a otros dominios. ELIZA ha servido de base a numerosos sistemas de gestión de diálogos [Björk, S. 1998] aunque, a pesar de su éxito, sea considerado por muchos de inferior calidad con respecto a los diálogos producidos por PARRY.

Sin embargo, aún después de estos logros, todavía a finales de la década de los ochenta la comunidad científica se cuestionaba si las reglas obtenidas del análisis de las conversaciones podrían ser eficazmente traspasadas a la lingüística computacional y, más allá, si los ordenadores podrían participar en conversaciones [Chapman, D. 1992]. Desde esta fecha paulatinamente ha aumentado el número de centros empresariales y de investigación que se dedica a la exploración y desarrollo de sistemas con interfaces de lenguaje hablado.

Cada vez es mayor la cantidad de desarrolladores y herramientas que pretenden conseguir que los sistemas de lenguaje hablado se conviertan en habituales, intentando incluso que principiantes sean capaces de crearlos de forma sencilla y automática [Sutton, S. and Cole, R. 1997]. De los primeros sistemas que incluían diálogos, que eran excesivamente rudimentarios y sólo permitían una interacción mediante comandos fijos, se ha pasado a los sistemas actuales que, ayudados por la madurez que ha alcanzado la tecnología, buscan poder procesar satisfactoriamente expresiones de habla no restringidas.

2.9.2 Criterios y consideraciones iniciales en el desarrollo de gestores de diálogo

Esta evolución ha provocado que los gestores de diálogos sean cada vez más sofisticados. Según se incrementa el número de tareas que se han de controlar mediante los diálogos, y según crece el campo de control de estas tareas, aumenta la complejidad del gestor de diálogos que se ha de construir. Por lo tanto, conviene predecir la complejidad del nuevo sistema que se va a desarrollar [Zadrozny, W. 1996] utilizando, por ejemplo, métodos de ingeniería del software.

Uno de los aspectos fundamentales que se debe tener en cuenta durante el desarrollo de un sistema de gestión de diálogos es la satisfacción final del usuario al utilizar el mismo. Conviene asegurarse con anterioridad de que los diálogos elegidos se adapten a las tareas. Haciendo un análisis de las tareas que desarrollará el sistema se puede revisar y refinar el modelo de diálogo para que se adapte de forma óptima.

Este tipo de análisis se puede realizar mediante estudios de simulaciones *Mago de Oz* (Wizard-of-Oz simulation) [Bretan, I. et al. 1995] que permiten a los desarrolladores obtener diálogos óptimos respecto a la naturalidad y eficacia hacia el usuario final. Estas simulaciones consisten en que una persona, y no una máquina, interprete las oraciones del usuario y actúe en consecuencia. El usuario al utilizar el sistema creerá que está interactuando con un sistema de gestión de diálogos, aunque realmente sus oraciones son procesadas por una persona [Dahlbäck, N. et al. 1993]. Este mecanismo también permite utilizar tecnología que todavía no esté disponible, como por ejemplo comprensión de lenguaje natural sin restricciones. Las simulaciones *Mago de Oz* se emplean como método para examinar y probar sistemas de diálogos de forma extensa desde principios de los ochenta, aunque también es un método adecuado para otros tipos de interfaces inteligentes.

A la hora de escoger un modelo de diálogos adecuado a las tareas que se desean desempeñar, se ha de considerar el rendimiento del gestor de diálogos desarrollado. Este rendimiento se determina sopesando el éxito en la tarea y el coste basado en el diálogo [Walker, M.A. et al. 1997]:

- Éxito en la tarea. Se mide teniendo en cuenta hasta qué punto el sistema y el usuario cumplen los requisitos de la tarea una vez que ésta ha concluido.
- Coste del diálogo. Se puede determinar mediante el número de turnos de diálogo necesarios y tiempo transcurrido para completar la tarea.

2.9.2.1 Iniciativa en los diálogos

Durante el desarrollo de un gestor de diálogos resulta imprescindible elegir cuidadosamente un modelo de diálogos adecuado. Para elegir un modelo que se adapte a las características del sistema que se está desarrollando resulta fundamental determinar modelos computacionales que definan cómo se controla la iniciativa en un diálogo.

Dentro de las interfaces gráficas de usuario (GUI) la mayor parte de las interacciones corresponden a iniciativas humanas. Sin embargo, dentro de las interfaces de usuario orales (SUI) el modelo más utilizado fue inicialmente el de iniciativa por parte del sistema (SI, System Initiative). En este modelo se guía al usuario sobre qué puede decir en cada momento con instrucciones directas. El modelo SI ha ido evolucionando y cada vez se presenta un mayor número de sistemas que utilizan un estilo de diálogo de iniciativa mixta (MI, Mixed Initiative). En el modelo MI se asume que el usuario sabe qué puede decir dado su actual contexto de interacción y por lo tanto no es necesario ofrecerle información al respecto.

Los gestores de diálogo se pueden clasificar dependiendo de qué estrategia de iniciativa utilicen. Tal y como se detalla en el estudio realizado por [Walker, M.A. et al. 1998], la estrategia de la MI resulta más eficiente (considerando el número de turnos necesarios para completar una tarea y el tiempo transcurrido) que la estrategia SI. Aunque ésta siempre está condicionada al reconocedor ya que su rendimiento es sensiblemente menor en un sistema MI debido a los problemas que presenta para los usuarios saber qué es lo que pueden decir. Esta tasa de reconocimiento va en aumento a medida que los usuarios se familiarizan con el sistema, esto es, a medida que los usuarios aprenden cómo pueden interaccionar con el mismo.

Considerando estas conclusiones, se podría suponer que los usuarios preferirían utilizar inicialmente un sistema de estrategia SI pero que a medida que éstos fueran ganando experiencia acabarían decantándose por un sistema con estrategia MI. Sin embargo, tal y como demuestra [Walker, M.A. et al. 1998], a pesar de que la experiencia del usuario y el rendimiento del reconocedor aumentan al utilizar el sistema, la flexibilidad que supone una interfaz MI puede generar cierta confusión, por lo que las personas muchas veces acaban prefiriendo una interfaz de estilo SI. Este estudio coincide con el de [Guinn, C.I. 1999], que detalla que los principales factores que contribuyen a la satisfacción del usuario son:

- Percepción por parte del usuario de que se ha completado con éxito la tarea desarrollada.
- Tiempo transcurrido para que esto ocurra.
- Rendimiento del reconocedor.

De entre estos tres factores, el último resulta mucho más importante que los dos anteriores, por lo que en ambos estudios se concluye que las preferencias de los usuarios se decantan por un modelo SI, antes que por una interfaz MI, ya que el rendimiento del reconocedor, la previsibilidad del sistema y la capacidad de los usuarios para adquirir un modelo de su comportamiento resultan aspectos sustancialmente más importantes que la consecución de la tarea o la eficiencia.

Esta no es la única clasificación posible si se tiene en cuenta la iniciativa dentro de los diálogos. También se puede distinguir entre iniciativa orientada a tareas e iniciativa orientada a diálogos [Chu-Carroll, J. and Brown, M.K. 1997]:

- Iniciativa orientada a tareas. Aquella en la que las oraciones proponen directamente acciones que los agentes deben realizar.
- Iniciativa orientada a diálogos. En la que el objetivo es establecer creencias mutuas sobre el conocimiento del dominio o sobre la validez de una sugerencia.

Una vez elegidos los criterios que debe cumplir el gestor de diálogos se puede iniciar el desarrollo del mismo. Para poder crear un gestor de diálogos, en primer lugar se ha de conseguir la trascripción de la frase pronunciada por el usuario de la forma más precisa que sea posible. Debido a los problemas en el reconocimiento de voz (ver los apartados 2.7.2 y 2.8.2.3) resulta necesario realizar una corrección de lo obtenido por el

reconocedor mediante métodos estadísticos, análisis sintáctico, análisis semántico y/o uso del contexto [Allen, J.F. et al. 1996]. Además, según la idea de [Ward, K. and Novick, D.G. 1995], para desarrollar interfaces del habla robustas no es suficiente una aproximación centrada exclusivamente en el reconocimiento del habla, sino que es necesario integrar diferentes fuentes de información (entonación, pausas, contexto anterior, etc.) que permitan determinar e interpretar mejor las oraciones de los usuarios.

2.9.2.2 Lexicones y gramáticas de los gestores de diálogos

Para poder transcribir y realizar un correcto procesamiento de las oraciones pronunciadas por el usuario (tal y como ya se ha comentado en los apartados 2.7.2 y 2.8.3) estos sistemas necesitan tener un vocabulario (lexicon) que contenga todas las palabras que el sistema es capaz de reconocer y con las que se puede trabajar. El lexicon no tiene por qué ser único. Puede darse el caso de que exista un lexicon que se utilice por el reconocedor y otro diferente para la gestión de diálogos [Martin, P. et al. 1996].

Estos diccionarios determinan en gran medida la capacidad del sistema para entender lo expresado por el usuario y su habilidad para procesar esa información y generar una respuesta válida y diferente en cada caso. Las palabras almacenadas en los lexicones no tienen por qué permanecer de forma estática. Existen diferentes mecanismos para hacer que el sistema aprenda nuevas palabras de forma dinámica [Roy, D. and Pentland, A. 1998a] de modo que éste se pueda adaptar de forma más precisa a las características de los interlocutores.

Para una correcta gestión de diálogos se necesita la presencia de gramáticas de diálogo [Dahlbäck, N. and Jönsson, A 1992]. Estas gramáticas presentan la ventaja de poder representarse como gramáticas independientes del contexto, y por lo tanto su tiempo de procesamiento será polinomial.

La creación de estas gramáticas requiere ciertos conocimientos lingüísticos. Para paliar este problema son cada vez mayores los esfuerzos en el desarrollo de herramientas de creación automática de gramáticas mediante interfaces de alto nivel. Este es el caso de la creación de gramáticas mediante ejemplos [Lieberman, H. et al. 1998].

Se ha de buscar el equilibrio entre tener unas gramáticas lo suficientemente pequeñas que permitan una alta precisión en tiempo real y permitir al usuario que se exprese de la forma más libre como le sea posible. Tal y como se apunta en el apartado 2.7.2.2, un método para conseguir un correcto funcionamiento del reconocedor, consiguiendo una interacción natural no restringida con el sistema, es utilizar un conjunto de gramáticas que cambian de forma dinámica [Coen, M.H. et al. 1999b]. Además de un conjunto de gramáticas que permanecen siempre activas habrá otra serie de gramáticas que cambian dependiendo del contexto del entorno o de la aplicación.

2.9.2.3 Análisis de las oraciones

Una vez obtenida la frase que el usuario ha pronunciado se ha de determinar, con el mayor nivel de exactitud posible, el objetivo que pretende conseguir el usuario al pronunciarla. Según la idea de [Searle, J. 1969], que definió lo que se conocen como *Speech Acts* (ver apartado 2.9.4.1), todas las palabras que un usuario pronuncia tienen como fin último llevar a cabo acciones. Como veremos más adelante, esta teoría ha sido ampliamente aplicada, estudiada y considerada en posteriores trabajos y sistemas que implementan interfaces con gestores de diálogos.

Para analizar las oraciones se puede aplicar sobre el texto técnicas de procesamiento de lenguaje natural. Estas técnicas permiten al sistema aumentar considerablemente su capacidad para el entendimiento del lenguaje en sus múltiples formas. Su uso conlleva la utilización de analizadores sintácticos, analizadores semánticos, herramientas de resolución de la anáfora, etc.

El procesamiento de estas oraciones se puede realizar de forma paulatina en varios niveles de análisis [Horvitz, E. and Paek, T. 1999], refinando el conocimiento que se tiene sobre ellas, hasta extraer correctamente la información que se desea obtener. Aunque los métodos utilizados pueden variar dependiendo del sistema, una secuencia de proceso habitual es:

- Obtener las palabras que conforman la oración pronunciada por el usuario, realizar sobre ellas un análisis morfológico y consultarlas en los lexicones de que se disponga.
- Utilizando las gramáticas del sistema, se puede realizar un análisis sintáctico de la oración utilizando un *parser* para tal efecto.
- Resolver ambigüedades que hayan podido surgir durante el anterior análisis, utilizando información semántica.
- Una vez realizado un correcto análisis de la oración, utilizar métodos de resolución de la anáfora, de la elipsis, además de otros métodos, dependiendo de la aproximación llevada a cabo por el sistema, que permitan determinar de forma óptima el significado de la frase pronunciada por el usuario.

2.9.3 Aproximaciones a la gestión de diálogos

La gestión de diálogos se han clasificado tradicionalmente en gramáticas de diálogo (Dialogue grammars), aproximaciones basadas en planes (Plan-Based approaches) y aproximaciones colaborativas (Collaborative approaches) [Churcher, G. et al. 1997].

2.9.3.1 Gramáticas de diálogos

Una de las primeras aproximaciones a la representación del diálogo fue el uso de una gramática preceptiva para secuencias de oraciones en un diálogo. Las primeras gramáticas describían la estructura del diálogo completo mientras que aproximaciones

más recientes tienen en cuenta secuencias regulares que se repiten en los diálogos, denominadas pares de adyacencia.

Esta aproximación se utiliza para analizar la estructura de un diálogo mediante el empleo de reglas de estructura de las frases y de máquinas de estados finitos. Sin embargo, ha sido criticada debido a su falta de flexibilidad para desviarse en los diálogos y para aplicarse en otros dominios [Levinson, S.C. 1981].

2.9.3.2 Basadas en planes

Esta resulta una aproximación más compleja en el modelado de diálogos. Se basa en el modo en que los humanos se comunican para alcanzar varios objetivos, intentando catalogar estos objetivos y los subobjetivos que les abarcan. En este caso es responsabilidad del oyente identificar el plan subyacente de su interlocutor y responder de forma adecuada. Esta aproximación intenta modelar esta idea y representar de forma explícita los objetivos de la tarea.

Las críticas al sistema se basan en que, en el peor de los casos, los procesos de reconocimiento del plan y de planificación son combinatoriamente inmanejables. Además éste no captura cuál es la motivación de los participantes para hablar en un diálogo. Los planes se basan en la estructura de la tarea pero necesitan incorporar alguna forma de meta-nivel que muestre cómo se pueden manipular los planes cuando secuencias de clarificación o similares aparecen.

2.9.3.3 Colaborativas

Estas aproximaciones se basan en ver a los diálogos como un proceso colaborativo. Ambas partes trabajan juntas para alcanzar un entendimiento mutuo del diálogo. Estas motivaciones generan fenómenos en el discurso del tipo de confirmaciones y clarificaciones, que resultan comunes en las conversaciones entre humanos.

Las aproximaciones colaborativas intentan capturar las motivaciones detrás de un diálogo y los mecanismos del propio diálogo, en lugar de concentrarse en la estructura de la tarea. Para ello es necesario modelar explícitamente las creencias de al menos dos participantes. Un objetivo propuesto, que es aceptado por los otros interlocutores, se convierte en parte de la creencia compartida y todos trabajarán de forma conjunta para alcanzar este objetivo.

2.9.4 Codificación de esquemas en la gestión de diálogos

La codificación de esquemas se utiliza a menudo para modelar la conversación en modelos deterministas (por ejemplo en una máquina de estados finita) o no deterministas (como los estadísticos). La mayoría de estos esquemas se basan en el significado que el usuario quiere trasmitir en una oración. Esta relación entre lo que se pronuncia y la intención puede resultar difícil de procesar, por lo que se asiste a través del uso del contexto [Churcher, G.E. et al. 1997]. El trabajo en este sentido fue iniciado por [Austin, J.L. 1962] y desarrollado posteriormente por la teoría de los actos de habla

(speech acts) [Searle, J.R. 1969], de modo que muchos esquemas de codificación se basan en estas acciones del habla o un equivalente derivado. A continuación veremos esta teoría junto con alguna de las otras aproximaciones existentes.

2.9.4.1 Speech acts

Algunas oraciones se pueden considerar como si estuvieran "realizando un acto". No son afirmaciones sobre el mundo sino declaraciones que realizan una acción. Los actos de hablase pueden describir en:

- Actos locucionarios (locutionary acts). Los actos físicos que expresan una secuencia lingüística con un significado.
- Actos ilocucionarios (illocutionary acts). Los que realizamos al expresarnos, como preguntar, afirmar, etc.
- Actos perlocucionarios (perlocutionary acts). Las acciones que se realizan como efecto de las expresiones utilizadas.

Los actos de habla deberían satisfacer las llamadas condiciones de felicidad (felicity conditions). Estas condiciones se basan en determinar si la oración es tan efectiva como lo pretendía quien la pronunció. Además puede no existir una relación única entre una frase pronunciada y un acto. El hablante puede utilizar varias oraciones para realizar un solo acto de habla. En otras ocasiones una sola frase puede producir múltiples actos de habla.

Los actos ilocucionarios se pueden organizar en diferentes tipos. Por ejemplo [Searle, J.R. 1976] propone cinco tipos basados en el significado de los verbos, aunque también se pueden clasificar dependiendo del modo de comunicación [Davis, E. 1990] como actos declarativos, los que transmiten información; actos interrogativos, los que solicitan información; actos imperativos, los que realizan un requerimiento o emiten una orden; actos exclamativos, los que expresan una emoción y actos representativos, que dan pie a una condición.

Además de las taxonomías propuestas, en muchos sistemas se emplean actos específicos que dependen de la aplicación. Un contexto dado puede basarse en un pequeño subconjunto de actos de habla, de modo que en ocasiones se escogen los actos específicos que sean apropiados para el dominio de la aplicación. En otras ocasiones se crean nuevas taxonomías de actos de habla que son independientes del dominio en los niveles más altos pero se convierten en más dependientes del dominio según se va bajando de nivel.

Los actos de habla, al igual que otros actos, se utilizan con el propósito de alcanzar objetivos. Así se pueden definir en términos de condiciones previas que se deben cumplir antes de que se pueda llevar a cabo el acto y sus efectos.

2.9.4.2 Conversational games

La teoría de *Conversational games* [Kowtko, et al. 1992] añade una forma de estructura a un diálogo basado en *speech acts* generando secuencias bien formadas de actos locucionarios.

Considerando que un diálogo realiza una única tarea, se puede considerar al diálogo como una serie de transacciones, cada una de las cuales cumple una pequeña parte de la tarea, esto es, una subtarea. Cada transacción comprende uno o más juegos (games), donde cada juego es un par de actos de habla complementarios. Un juego consiste en un movimiento de inicio y (opcionalmente en ocasiones) un movimiento de finalización. Adicionalmente, un juego puede estar embebido en otro, permitiendo fenómenos del tipo de clarificación, desvíos, etc.

Así, se presenta como una herramienta para modelar la conversación persona-máquina en diálogos complejos orientados a tareas, de iniciativa mixta. Este sistema emplea técnicas de gramáticas de diálogos y de aproximaciones basadas en planes.

Existen otros intentos de derivar una estructura sobre el nivel de *speech acts* [Alexandersson, J. 1996]. Tomando un corpus de diálogos anotados con actos de diálogo se puede adquirir automáticamente una estructura intencional, esto es, una estructura jerárquica del diálogo, dividiendo el diálogo en objetivos y subobjetivos.

2.9.4.3 Information state

La teoría del estado de la información (Information State Theory) [Cooper, R. 1997], basada en la noción de Questions Under Discussion (QUD) de [Ginzburg, J. 1996], consiste en:

- Una descripción de los componentes informativos de la teoría (por ejemplo participantes, estructura lingüística e intencional, creencias, modelos de usuarios, etc.)
- Una representación formal de los componentes anteriores.
- Un conjunto de movimientos de diálogos que provocarán la actualización del estado de la información.
- Un conjunto de reglas de actualización, que controlan la actualización del estado de la información.
- Una estrategia de actualización para decidir qué reglas se seleccionan en un punto determinado de entre el conjunto de reglas aplicables.

El estado de la información se almacena de forma interna por un agente (por ejemplo, el sistema de diálogos). Se compone de una parte estática que incluye reglas para interpretar oraciones, actualizar la parte dinámica y seleccionar movimientos futuros y de una parte dinámica que varía dependiendo de los eventos que ocurran y de cómo se gestionen.

2.9.5 Sistemas de diálogos de ámbito general

A continuación veremos una breve introducción a algunos de los sistemas de diálogos más relevantes. Se describirán algunas de sus características más importantes y la forma utilizada para gestionar los diálogos.

2.9.5.1 Sundial

El objetivo del proyecto Sundial (Speech UNderstanding in DIALogue) [Peckham, J. 1993] fue construir sistemas de diálogos en tiempo real capaces de mantener diálogos cooperativos con usuarios a través de la línea telefónica [Peckham, J. 1991] en los dominios de consultas y reservas aéreas y de consultas sobre trenes. Los sistemas se desarrollaron en francés, inglés, italiano y alemán. La funcionalidad de los sistemas fue especificada parcialmente mediante simulaciones Mago de Oz y evaluada con usuarios potenciales bajo diversas condiciones.

Los sistemas realizan tres funciones principales: la interpretación de las oraciones de los usuarios, la generación de oraciones por parte del sistema y la gestión del diálogo de modo que las frases del sistema sean naturales y coherentes con las frases de los usuarios. La interpretación lingüística se realiza mediante un módulo de procesamiento acústico [Kuhn, T. et al. 1992] y un módulo de procesamiento lingüístico [Andry, F. and Thornton, S. 1991]. El módulo de gestión del diálogo toma la representación lingüística de la oración y le da una interpretación dentro del contexto del diálogo [McGlashan, S. et al. 1992]. Utilizando esta interpretación decide cómo podría continuar el diálogo y, si es el turno del sistema para hablar, planea una representación lingüística esquemática para la oración del sistema. La generación de la oración del sistema se realiza mediante un módulo de generación del mensaje y un módulo de síntesis del habla. Su modelo de aproximación se basa en gramáticas de diálogos (ver sección 2.9.3.1).

2.9.5.2 TRAINS

El proyecto TRAINS [Allen, F.J. et al. 1996] se ha desarrollado en la University of Rochester con los objetivos de emprender un amplio estudio de diálogos orales hombre-máquina, construir una serie de prototipos robustos con los que usuarios no entrenados pudieran interaccionar y utilizar estos prototipos como una plataforma que permita investigar en profundidad la comprensión del lenguaje natural, la planificación de iniciativa mixta y el razonamiento sobre el tiempo, las acciones y los eventos. Para ello se intentó llevar a la práctica las teorías existentes, en lugar de desarrollar nuevas teorías. El dominio del proyecto es un planificador de rutas, donde se debe encontrar la ruta en tren más eficiente entre dos destinos.

El gestor de diálogos es responsable de mantener el curso de la conversación y asegurarse de que se alcanzan los objetivos de ésta (el principal objetivo del sistema es ejecutar un plan que ha sido acordado por el usuario y el sistema y que se adhiere a las restricciones). Para hacer esto el gestor tiene que interpretar los actos de habla en

contexto, formular respuestas y mantener un modelo de su estado mental y del contexto del discurso. El modelo del estado mental incluye creencias anidadas y proposiciones sobre el dominio, un conjunto de objetivos actuales del discurso, un conjunto de actos de habla que el sistema pretende decir cuando tenga oportunidad y un conjunto de obligaciones de discurso pendientes [Allen, F.J. et al. 1995].

El sistema presta una atención especial a la capacidad de detectar e interpretar correcciones. En el caso de encontrar ambigüedades se considera mejor elegir una interpretación específica y correr el riesgo de cometer un error que generar un subdiálogo de clarificación. Su aproximación se basa en un modelo colaborativo (ver apartado 2.9.3.3).

Este proyecto ha evolucionado hacia el proyecto TRIPS [Ferguson, G. and Allen, J.F. 1998] donde se consideran nuevos medios de transporte, una mayor complejidad en la planificación, etc.

2.9.5.3 Trindi, Siridus y TrindiKit

El proyecto Trindi [Larsson, S. and Traum, D. 2000] propone una arquitectura y una herramienta para construir gestores de diálogos basados en la teoría *Information state* (ver apartado 2.9.4.3). El estado de la información de un diálogo representa la información necesaria para distinguirlo de otros diálogos, representando adiciones acumulativas de acciones previas del diálogo y motivando acciones futuras.

El objetivo es crear diálogos instructivos orientados a tareas, proponiendo la planificación de rutas como un escenario básico.

La gestión de diálogos y el trazado del discurso se realizan mediante un gestor de movimientos de diálogos (Dialogue Move Engine) que implementa la teoría de *Information State*. Sus principales funciones son actualizar el estado de la información basándose en las observaciones de movimientos y seleccionar los movimientos que se deben llevar a cabo.

Siguiendo el punto de vista de Trindi se ha desarrollado el proyecto Siridus dentro del V Programa Marco de la Unión Europea. Su objetivo es expandir la noción y desarrollar herramientas computacionales que permitan el desarrollo de sistemas de diálogos más robustos, funcionales y amigables. Para ello extienden el rango de los tipos de diálogos a los que se puede aplicar la aproximación de *Information State* [Amores, J.G.; Quesada, J.F. 2001].

TrindiKit [Larsson, S. and Traum, D. 2000] es una herramienta para construir y experimentar con *dialogue move engines* e *information states* desarrollada dentro de los proyectos Trindi y Siridus. Además de proponer la arquitectura general del sistema la herramienta también especifica formatos para definir estados de información, actualización de reglas, movimientos de diálogos y algoritmos asociados.

2.9.5.4 GALAXY

El sistema GALAXY [Seneff, S. et al. 1998] es un proyecto del Spoken Language Systems group del MIT para desarrollar una interfaz en lenguaje oral para información en línea. Se trata de un sistema: distribuido, que emplea una arquitectura cliente-servidor; multi-dominio, de modo que pueda proporcionar acceso a una amplia variedad de fuentes de información y dominios y extensible, permitiendo añadir nuevos servidores de dominio al sistema de forma incremental.

La arquitectura cliente-servidor hace que el sistema envíe respuestas generadas por los servidores a clientes sencillos (como ordenadores personales o teléfonos). Los servidores de tecnología proporcionan reconocimiento del habla [Glass, J. et al. 1996], comprensión del lenguaje [Seneff, S. 1992], gestión del diálogo [Seneff, S. and Polifroni, J. 2000], generación de lenguaje [Baptist, L. and Seneff, S. 2000] y síntesis de voz [Yi, J. et al. 2000]. Los servidores de dominio envían información específica que incluye información sobre el tiempo [Zue, V. et al. 2000], horarios y precios de vuelos [Seneff, S. and Polifroni, J. 2000], estado y puerta de acceso a vuelos y condiciones del tráfico y asistencia en la ciudad de Boston, entre otros.

El módulo de comprensión del lenguaje analiza las oraciones pronunciadas por los usuarios para obtener sus componentes gramaticales (como sujeto, verbo, objeto, predicado). Entonces amplia los componentes sintácticos con información semántica y convierte las oraciones en marcos semánticos: una estructura de tipo comando que contiene cláusulas, asuntos y predicados.

El gestor del diálogo es el que tiene que evaluar la relevancia y completitud del requerimiento del usuario, recuperar la información solicitada de la base de datos y establecer una respuesta adecuada en forma de un marco semántico.

El generador del lenguaje procesa los componentes del marco semántico y genera una representación en texto de las semánticas en el lenguaje requerido, ya sea un lenguaje natural como inglés o chino o un lenguaje formal como SQL (Structured Query Language).

El componente fundamental de la arquitectura es un *Hub* central programable que controla el flujo de datos entre los clientes y los servidores y mantiene el estado y la historia de la conversación actual. El sistema se distribuye bajo el nombre de Galaxy Communicator como una herramienta de uso libre que proporciona una arquitectura para construir sistemas de diálogos.

2.9.6 Sistemas de diálogos en entornos inteligentes

Como se ha mencionado a lo largo de este capítulo, la mayoría de los entornos inteligentes existentes no han explorado en profundidad las posibilidades de interacción mediante diálogos orales, sino que han dado más peso a otras modalidades de percepción tales como pequeños sensores (The Adaptive House, House_n, etc.) o

videocámaras (Aire, The Aware Home, The KidsRoom, etc.). En otras ocasiones el sonido se ha utilizado con otros fines.

Este el caso de The Aware Home. Un ejemplo es el estudio de las posibilidades del sonido para mejorar el aprendizaje en el empleo de dispositivos en adultos de diferentes generaciones [McLaughlin, A.C. et al. 2003]. En una posición diferente se han empleado micrófonos para realizar localización de los usuarios del entorno o reconocimiento de los individuos [Stillman, S. and Essa, I. 2001]. Por último, dentro de The Aware Home, el audio también se ha utilizado en un sistema de recuperación de información en conversaciones grabadas.

En una vertiente similar House_n utiliza el sonido para aplicaciones diversas como dejar y reproducir anotaciones y mensajes de audio en el entorno o emplear capturas con cámaras y micrófonos para deducir la actividad que se está realizando [Intille, S.S. et al. 2003].

Dentro del proyecto EasyLiving se han analizado las diferentes cuestiones que surgen en el proceso de interacción con los entornos inteligentes [Shafer, S.A.N. et al. 2001] e incluso se ha conducido un estudio que determina que la interacción oral es fundamental como medio de interacción en el hogar [Brumitt, B. and Cadiz, J.J. 2001]. A pesar de esto no hemos encontrado publicaciones donde se describa algún sistema desarrollado de interacción oral con su entorno.

En cuanto a la iRoom de Stanford, como ya se ha dicho, centra sus interacciones en el acceso a pantallas de gran formato. Dentro de este concepto han desarrollado un auditorio virtual que sirve como sistema de videoconferencia para el aprendizaje a distancia. El instructor tiene acceso a docenas de estudiantes a la vez, pudiendo establecer contacto visual con ellos. La información es transmitida utilizando audio y vídeo [Chen, M. 2001].

Por otro lado The Adaptive House siempre ha trabajado con la premisa de que su entorno inteligente no debe tener ningún tipo de interfaz (oral o de cualquier otro modo) que se aleje de los controles habituales que se encuentran en el hogar [Mozer, M.C. 2004].

Aún así, existen algunos sistemas de diálogos orales para la interacción con el entorno de los que resulta interesante realizar un análisis.

2.9.6.1 Intelligent room, HAL y Aire

Desde el inicio del desarrollo de estos proyectos, dentro del MIT Artificial Intelligence Laboratory consideraron necesario que su entorno inteligente pudiera llevar a cabo interacción orales en lenguaje natural [Coen, M.H. 1998].

La principal característica del sistema de diálogos desarrollado es el bosque de reconocimiento, una estructura de datos lingüísticos cuyo objetivo es simplificar la incorporación de información de contexto en el sistema de comprensión del habla y

permitir que múltiples aplicaciones del entorno accedan a esta modalidad de forma independiente [Coen, M. et al. 1999b].

Este sistema de diálogos está pensado para no monopolizar la atención del usuario o requerir un conocimiento excesivo de las palabras necesarias para interaccionar. En cambio se pone más énfasis en realizar reconocimiento del habla en entornos ruidosos, donde conviven múltiples usuarios. El sistema se aleja de una aproximación dependiente de la tarea o el dominio, al ser una herramienta general que puede ser utilizada de forma simultánea por diversas aplicaciones.

Los ocupantes del entorno portan micrófonos de solapa que transmiten la información reconocida al sistema de comprensión del habla. Por defecto la habitación ignora las oraciones pronunciadas por sus habitantes, ya que estas se suelen dirigir a otros ocupantes del entorno. Cuando un usuario quiere atraer la atención del entorno, tras una breve pausa, debe pronunciar la palabra *Ordenador*. En ese momento el sistema emite un ligero pitido (ver apartado 2.8.1) para indicar que está prestando atención. En ese momento el usuario dispondrá de una ventana de dos segundos para realizar una interacción con la habitación. Una vez finalizado el tiempo o la interacción el sistema vuelve a dormir.

El sistema atiende a las oraciones pronunciadas por los usuarios que están contenidas en un bosque (conjunto) de múltiples gramáticas. Cada gramática en este bosque de reconocimiento se crea por uno de los agentes software de la habitación, que reciben una notificación cuando una oración contenida en una de sus gramáticas ha sido reconocida. El mensaje de notificación de una frase reconocida contiene un árbol gramático que el agente puede manipular para determinar su contenido.

En este bosque de reconocimiento una gramática se considera activa si la habitación está en ese momento considerando las oraciones que contiene, e inactiva en el caso contrario. Asumiendo que ciertos tipos de oraciones sólo tienen probabilidad de pronunciarse bajo ciertas circunstancias, los agentes software de la habitación tienen la responsabilidad de modificar los estados de activación de las gramáticas que ellos crean, basándose en la información que proporciona el entorno u otros agentes software. A su vez, las gramáticas activas se ordenan según la probabilidad en que se estima serán oídas. Cuando un usuario cambia a un nuevo contexto de aplicación el sistema disminuye el peso relativo o incluso desactiva las gramáticas de contextos previos.

Además de para interaccionar con los elementos del sistema, el habla se ha utilizado en estos proyectos con diferentes fines. Un par de ejemplos son:

- Una interfaz para programar entornos inteligentes mediante lenguaje natural [Gajos, K. et al. 2002]. Esta aplicación programa el sistema grabando "macros orales" donde el usuario registra el nombre de un nuevo objetivo, demuestra uno o más planes para alcanzar ese objetivo y las condiciones en que preferiría un plan sobre el otro (por ejemplo, encender las luces si un usuario entra en la habitación pasada una determinada hora).

- Una aplicación que mezcla dibujos de bocetos con explicaciones orales de los dibujos para obtener una interfaz más natural en un entorno de diseño [Adler, A. and Davis, R. 2004]

2.9.6.2 D'Homme

El proyecto D'Homme, Dialogues in the Home Machine Environment, financiado por el V programa marco de la Unión Europea afronta los retos teóricos que surgen en la comprensión del lenguaje y la gestión de los diálogos para controlar y consultar el estado de los dispositivos del hogar. Entre los participantes del proyecto se encuentran las compañías SRI International y Netdecisions y las universidades de Göteborg, Edinburgh y Sevilla, adaptando cada uno de sus sistemas de diálogos a la interacción con el hogar.

Entre los resultados del proyecto se puede destacar:

- La obtención de una arquitectura para un sistema de diálogos orales que admite el uso de dispositivos *plug and play* [Rayner, M. et al. 2001]. Cada dispositivo en el entorno del hogar lleva información lingüística y de gestión de diálogos relacionada con el mismo y que se carga dinámicamente a los componentes de procesamiento de lenguaje relevantes en la interfaz oral. Con estas premisas se ha realizado un demostrador con dispositivos que permiten acciones de encendido y apagado, de regulación y con sensores. Además, emplean gramáticas de unificación [Pullum, G.K. and Gazdar, G. 1982] con el objetivo de adecuarse con más precisiones a las posibles interacciones con los usuarios y mejorar la tasa de reconocimiento.

- Un estudio sobre las ventajas de lenguajes de modelos estadísticos y de sistemas basados en gramáticas en el reconocimiento en el dominio del control de dispositivos en el hogar [Knight, S. et al. 2001]. Como resultados se obtuvo que los sistemas basados en gramáticas son mejores para usuarios con experiencia en cómo interaccionar con el entorno y que, por el contrario, los modelos estadísticos ofrecen mejores resultados con usuarios noveles, apuntando la posibilidad de utilizar una conjunción de ambos sistemas.

- Una arquitectura basada en agentes para el diseño e implementación de un sistema de diálogos orales para el control de las luces en el hogar [Quesada, J.F. et al. 2001]. El sistema se basa en movimientos de diálogos para lenguajes de comandos naturales [Amores, J.G.; Quesada, J.F. 2001] y gramáticas de unificación. Es capaz de resolver diferentes fenómenos, como cuantificación, resolución de dispositivos, reparación de errores, resolución de la anáfora, coordinación, etc. [Quesada, J.F. and Amores J.G. 2002].

Una extensión a este proyecto viene dada por el sistema de diálogos de Linguamatics que crea un sistema de diálogos genérico para el control de los dispositivos del hogar basándose en conocimiento ontológico del dominio [Milward, D. and Beveridge, M. 2003].

Los estudios realizados en este proyecto se basan en simulaciones de control de dispositivos en el entorno del hogar. En este caso no se ha producido una sinergia de entornos inteligentes reales con interfaces orales, sino que centran su atención en estudiar y experimentar con sistemas y arquitecturas de diálogos adecuadas para el hogar.

2.9.6.3 Aproximaciones multimodales: Smartkom y Embassi

Las interacciones multimodales, aunque todavía presentan numerosas limitaciones, constituyen una gran ventaja ya que aumentan la naturalidad de la interacción y permiten resolver conflictos que aparecen con una única modalidad. A continuación describimos un par de sistemas multimodales desarrollados para interaccionar con el entorno.

SmartKom

El proyecto Smartkom se despega de la mayor parte de los proyectos vistos hasta ahora para proporcionar una interfaz auténticamente multimodal. Su propuesta ha sido desarrollar una interfaz persona-ordenador que sea intuitiva de utilizar y se adapte a las necesidades y preferencias de los usuarios. El sistema reconoce habla, gestos y señales y genera texto, gráficos y habla. Los usuarios y el sistema pueden utilizar cualquier modalidad que se considere más apropiada para la tarea particular, el tipo de información, las preferencias del usuario o el escenario de aplicación. Existen tres posibles escenarios: un entorno de trabajo en el hogar o la oficina donde se proporcionan servicios de información multimodales (una guía de programación para televisión, control de dispositivos electrónicos, reproductores de vídeo, etc.), puntos de acceso públicos a servicios de información (un quiosco de comunicación para aeropuertos, estaciones, etc. donde se puede buscar información sobre hoteles, restaurantes, etc.) y dispositivos móviles (PDAs que permiten navegación por ciudades, etc.) [Wahlster, W. et al.]. El proyecto ha sido desarrollado dentro del programa *Human Computer Interaction*, financiado por el ministerio alemán de educación e investigación, por un consorcio académico e industrial liderado por el centro de investigación alemán de inteligencia artificial (DFKI) y es una continuación del proyecto Verbmobil cuyo objetivo era desarrollar un sistema de traducción móvil para la traducción de habla espontánea en situaciones cara a cara [Görz, G. et al. 1999].

Una de las hipótesis fundamentales de SmartKom es que el usuario obtiene una interacción homogénea y agradable a través de un agente de interacción antropomórfico personalizado, a quien se delega la tarea que se debe resolver en los tres escenarios. Ambas partes en la comunicación colaboran durante el proceso de resolución del problema en el cual el agente personalizado accede a los servicios. Este agente puede solicitar más información y finalmente presentar los resultados en alguno de los canales de salida [Reighinger, N. et al. 2003].

Las modalidades de las que dispone el usuario son fundamentalmente habla, procesada por un reconocedor del habla independiente del usuario y gestos, reconocidos por un hardware específico del escenario. Para todas las entradas y salidas

multimodales se emplea una representación común que permite utilizar modelos de interacción genéricos, basada en XML lo que asegura una sintaxis clara y bien definida para el intercambio y la validación de datos. El esquema principal, describiendo las intenciones del usuario y del sistema, se define en una ontología de modo que todos los asuntos sobre los que el usuario y el sistema pueden hablar están codificados en la misma. El sistema está basado en una plataforma de integración con una arquitectura de componentes distribuida. La plataforma está implementada sobre la base de una aproximación de publicación/suscripción. Los módulos software se comunican mediante repositorios de datos que corresponden con colas de mensajes.

La información multimodal se fusiona para conseguir una interacción más natural y para asistir a los posibles fallos que se produzcan en el reconocimiento de voz. La salida del intérprete de lenguaje es una lista de hipótesis de las intenciones del usuario que contienen representaciones ontológicas de la entrada, marcas de tiempo y de probabilidad. Lo mismo ocurre con la salida del analizador de gestos. Su fusión combina ambas listas y con la ayuda de la memoria del diálogo se obtiene la lista de las posibles hipótesis. La memoria del diálogo es un repositorio centralizado donde se representa información sobre el mismo [Pfleger, P. et al. 2003]. Además de permitir realizar resolución de referencias su principal tarea consiste en combinar nueva información con conocimiento previo.

Una vez obtenidas las nuevas hipótesis la lista de intenciones se pasa al planificador de acciones. Su tarea es coordinar las acciones que tiene que llevar a cabo el sistema de diálogos. Este planificador tiene que seleccionar la aplicación adecuada según el requerimiento del usuario y enviar el contenido que debe ser mostrado por el módulo de presentación.

El módulo de presentación decide cómo se van a mostrar los contenidos basándose en diversas condiciones que van desde una distribución apropiada a través de las modalidades de salida a restricciones sobre las modalidades disponibles. Un modelador de interacción proporciona la información sobre las modalidades preferidas y disponibles, analizando el escenario y las preferencias específicas del usuario.

Embassi

Embassi es otro proyecto del ministerio alemán de educación e investigación en el que participan diecinueve socios del mundo académico e industrial. Su propósito es hacer más sencilla e intuitiva la interacción con infraestructuras técnicas de uso cotidiano, tales como equipos de control y de entretenimiento en el hogar, terminales públicas o sistemas multimedia de los automóviles, mediante una aproximación multimodal.

Un componente de fusión de modalidades une habla, vídeo, gestos de señalado y la entrada de una interfaz gráfica. Un componente de planificación de la presentación decide qué modalidad se va a utilizar como salida entre habla, un carácter animado y/o una interfaz gráfica y se asegura que la presentación es coherente y consecuente.

Su planteamiento y arquitectura [Elting, C. et al. 2003] resulta muy similar al de SmartKom. Los análisis de los componentes de todas las modalidades generan salidas en un formato de descripción de la semántica común, con el fin de poder añadir o eliminar analizadores de modalidad de forma sencilla y dinámica. La entrada está gestionada por módulo de entrada polimodal que combina las modalidades de entrada y la salida por un módulo de salida polimodal que decide sobre la modalidad de salida. La arquitectura se compone de numerosos agentes agrupados en capas que gestionan la información a diferentes niveles de abstracción. Los agentes se comunican mediante el lenguaje de comunicación de agentes KQML (Knowledge Query and Manipulation Language) [Finin, T. et al. 1994] con una sintaxis expresada en XML conforme a un DTD (Document Type Declaration) que describe la ontología subyacente utilizada en el sistema. Los agentes pueden requerir la ayuda de otros agentes para completar una tarea.

2.10 Conclusiones sobre los sistemas de diálogos en entornos inteligentes

La incorporación de un sistema de diálogos orales puede resultar un componente fundamental en el desarrollo de un entorno inteligente. Sin embargo son todavía varios los problemas que se deben afrontar para poder realizar esta tarea con éxito. De entre ellos se pueden destacar dos que son de especial relevancia:

- El primero es común a la inmensa mayoría de los sistemas de diálogos orales aunque se ve acrecentado por las características heterogéneas, móviles y ruidosas de los entornos inteligentes: los fallos en el reconocimiento de voz.
- El segundo de ellos se produce debido a las características dinámicas de los entornos inteligentes. Los diálogos han de adaptarse de forma automática a entornos diferentes, e incluso cambiar considerablemente a lo largo de la vida del entorno.

Los fallos en el reconocimiento de voz pueden arruinar cualquier interfaz de diálogos orales. De ahí que resulte esencial aplicar mecanismos efectivos de disminución y recuperación de fallos. Para esta tarea se pueden considerar tres aproximaciones posibles: calibración del micrófono, gestión del diálogo y empleo de diccionarios y gramáticas.

La elección de un tipo micrófono (o entrada de sonido) adecuado puede resultar de gran ayuda para mejorar la tasa de reconocimiento. Sin embargo, las características de la interfaz no siempre permiten emplear la solución más apropiada. Un micrófono de proximidad con cancelación de ruidos puede resultar una opción conveniente aunque sin embargo no constituye la mejor solución para un espacio móvil como el de los entornos inteligentes. Por el contrario los micrófonos de ambiente pueden resultar inadecuados en un entorno con fuentes de sonido móviles y altos niveles de ruido.

Algo similar ocurre con el entrenamiento del reconocedor. En un entorno abierto que puede ser utilizado por numerosas personas (donde además puede resultar imposible determinar o prever su identidad) esta opción puede llegar a ser inviable.

También es importante realizar las elecciones correctas en la gestión del diálogo. En primer lugar se han de considerar las características que ha de tener el sistema y el tipo de usuarios al que va dirigido. Para ello conviene hacer un estudio previo en el que se pueden utilizar, por ejemplo, técnicas del *Mago de Oz*. Un aspecto crucial consiste en establecer un modelo de iniciativa adecuado. De esta forma se consigue que el diálogo oriente al usuario a través de la tarea que quiere realizar, evitando que este se sienta desconcertado ante la capacidad del gestor de diálogos o su comportamiento.

Una última aproximación orientada a mejorar las tasas de reconocimiento y comprensión consiste en el empleo de diccionarios y gramáticas específicas. Si bien su empleo reduce la cantidad de posibles interacciones con el sistema su correcta configuración puede mejorar la recuperación de errores sin restringir de forma notable la libertad de interacción del usuario. Un inconveniente que se debe considerar en el uso de esta técnica dentro de los entornos inteligentes viene dado por sus características dinámicas. Las gramáticas deben adaptarse a cada entorno distinto además de al estado en que se encuentran en cada momento.

Este dinamismo de los entornos también hace que la interfaz de diálogos orales deba ser capaz de adaptarse a los cambios. En estos espacios los sistemas de diálogos nos pueden considerarse como algo estático sino que han de adaptarse desde el primer momento al entorno con que trabajan. El esfuerzo necesario para crear o adaptar estas interfaces dentro de nuevos entornos debe ser nulo o mínimo. A su vez las interfaces han de adaptarse fácilmente a los diferentes estados y elementos que pueden componer el entorno a lo largo del tiempo.

Siguiendo con esta idea resulta fundamental dentro de cualquier interfaz de diálogos orales, pero muy especialmente en las que estén relacionadas con entornos inteligentes, considerar el contexto no sólo del diálogo sino del espacio en el que se desarrolla. Esta información de contexto puede ayudar tanto a recuperarse de errores y hacer una correcta interpretación de las oraciones como a ofrecer respuestas más apropiadas y consecuentes.

Otra posible aproximación que sirve como apoyo al reconocimiento e interpretación de las oraciones (y que también se utiliza para incrementar la naturalidad en la interacción) consiste en el empleo de técnicas multimodales. Este tipo de técnicas parecen convenientes dentro de este tipo de entornos ya que se aproximan en mayor medida al modo de comunicación entre humanos. El reconocimiento de gestos, de escritura o de señales puede resultar un muy buen complemento a la información recibida de forma oral, bien sea para resolver información ambigua o como sustituto a otras modalidades de información.

3 Propuesta de entorno inteligente

"Cada uno es como Dios lo hizo, y aun peor muchas veces
Sancho – Capítulo IV – Segunda Parte"

Dentro de la investigación realizada en esta tesis se ha desarrollado un entorno inteligente real que cumple dos funciones primordiales. En primer lugar trasladar al mundo físico las ideas desarrolladas sobre entornos inteligentes. En segundo lugar servir como banco de pruebas para el desarrollo de aplicaciones e interfaces que interactúan con estos entornos.

Las principales características del entorno inteligente que se ha desarrollado son:

- El control de los dispositivos físicos del entorno se realiza mediante el bus domótico EIB (European Installation Bus).
- Para el flujo de información multimedia se emplea una red Ethernet.
- La definición del entorno y los elementos que forman parte de él se realiza de forma estándar y sencilla, basándose en el lenguaje XML.
- A partir de esta definición se crea un repositorio común, que se denomina pizarra, donde queda representado el entorno, sus elementos, relaciones, estado, flujos de información, aplicaciones, etc.
- La información física de los dispositivos del entorno se encuentra encapsulada en esta capa de pizarra, de tal modo que se puedan cambiar unos dispositivos por otros de igual funcionalidad sin variar las capas superiores.
- Este repositorio almacena información de contexto que puede ser utilizada por el resto de aplicaciones.
- Esta información se encuentra continuamente actualizada de acuerdo con la situación actual del entorno.
- Para modificar el estado de alguno de los elementos del entorno bastará con enviar mensajes estándar de alto nivel a la pizarra. A su vez, para obtener el estado de algún elemento, basta con realizar una solicitud a la misma.

El entorno desarrollado se compone de un conjunto de dispositivos físicos de control (sensores y actuadores), dispositivos multimedia, aplicaciones software de alto nivel e interfaces de usuario (ver apartado 3.5). Las aplicaciones emplean la información de contexto almacenada en la pizarra para realizar acciones y las interfaces permiten una interacción natural con el entorno.

3.1 Capa física

Los elementos físicos que se pueden controlar en un entorno inteligente se pueden clasificar en dispositivos de control y dispositivos multimedia. Entre los dispositivos de control se encuentran los sensores (de presencia, temperatura, etc.) y los actuadores (en luces, puertas, electrodomésticos, etc.). Los dispositivos multimedia engloban a radios, televisores, pantallas de información, micrófonos, altavoces, etc.

Los dispositivos de control se gestionan mediante el bus europeo EIB (actualmente denominado KNX). Este bus surge como resultado del proceso de estandarización en el

control de los elementos de una red domótica. Los dispositivos presentan una arquitectura distribuida. Cada elemento lleva incorporado de forma implícita la información de procesamiento que necesita en microcontroladores, de modo que cada uno funciona de forma independiente a los demás y es el bus el que se ocupa de comunicarlos.

Los dispositivos multimedia están conectados a una red Ethernet. Estas redes resultan estándar, conocidas y ampliamente utilizadas dentro del campo de las telecomunicaciones, por lo que su utilización simplifica el empleo y desarrollo de aplicaciones multimedia dentro del entorno. Ambas redes se armonizan mediante una capa SMNP (Simple Management Network Protocol) [Martínez, A.E. et al. 2003].

Los dispositivos físicos se comunican con una capa *middleware* (ver más adelante el apartado 3.3) utilizando mensajes estándar en XML y el protocolo HTTP (HyperText Transfer Protocol). Por lo tanto, basta con que la entidad (ya sea un dispositivo físico o una aplicación) que vaya a formar parte del entorno sea capaz de recibir estos mensajes utilizando el protocolo HTTP.

Las entidades disponen de la API de comunicación proporcionada por el sistema. Esta API permite la interacción de los dispositivos con el *middleware*. A su vez, nuevos dispositivos pueden crear nuevas APIs para lenguajes específicos. En todo caso, el protocolo y los mensajes son estándar y abiertos, de modo que resulta sencillo desarrollar nuevas APIs de comunicación.

3.2 Lenguaje de descripción del entorno

Dada las características dinámicas y heterogéneas de los entornos inteligentes a las que ya se ha hecho mención previamente (ver capítulo 2) resulta de gran interés disponer de un mecanismo de descripción del entorno, donde se puedan definir de forma armónica los elementos que lo componen y sus características. Esta descripción puede servir, a su vez, como una herramienta de *documentación* del entorno, de modo que pueda ser utilizada por aquellas personas que trabajen desarrollando aplicaciones para el entorno inteligente como un documento en donde quedan plasmados qué elementos se encuentran presentes, cuáles son los atributos que tienen disponibles y, por lo tanto, cuáles son las posibles formas de interacción con éstos.

Con tal fin se ha creado un lenguaje de descripción (DL) del entorno basado en XML. Mediante este leguaje además de establecer qué dispositivos físicos se encuentran en el entorno se pueden definir aplicaciones software, personas o conceptos abstractos, así como las relaciones entre todos ellos. Las relaciones permiten definir la distribución del entorno (en edificios, habitaciones, etc.), los grupos de personas (por grupos de trabajo, rango, parentesco, etc.) o enlaces dinámicos entre elementos (los elementos preferentes para una persona, el altavoz de salida para una fuente de sonido, etc.).

Los elementos del entorno se denominan entidades. Cada entidad posee una colección de propiedades. Las entidades del mismo tipo heredan un conjunto de propiedades comunes, que definen sus características específicas, y que tienen un nombre único para

cada entidad y un valor. A su vez, las instancias de entidades pueden definir nuevas propiedades comunes, denominadas parámetros, que representan información específica para esa entidad. Los parámetros se agrupan en conjuntos que definen características relacionadas de la entidad.

3.2.1 Definición de las clases de entidad

Los tipos posibles de entidad se definen en uno o varios documentos de descripción de las clases de entidad (DDCE). Cada clase de entidad define las características comunes que tiene toda entidad de ese tipo. Las instancias de esa clase de entidad heredan las características comunes de la clase y pueden, en caso de ser necesario, añadir propiedades específicas de la instancia (ver siguiente apartado).

De este modo, cuando se diseña un nuevo tipo de dispositivo físico o de aplicación en el entorno se acompaña de un DDCE, que corresponde con la documentación en XML de la descripción de las características comunes de ese tipo de entidad (ver apartado 5.4).

Cada clase se define mediante sus propiedades (especificadas mediante la etiqueta *property*) y, opcionalmente, un conjunto de parámetros comunes (representado con las etiquetas *paramSet* y *param*). Los parámetros pueden estar asociados a una propiedad concreta o a toda la clase. Además, cada clase hereda las propiedades de uno de los tipos al que pertenece (mediante el atributo *extends*). Estos tipos definen características comunes. La Figura 1 muestra la sintaxis de un DDCE.

```
<classes>
   definición_clase
   [, definición_clase ] ...
</classes>

definición_clase:
<class name="nombre" extends="tipo">

   <property name="nombre_propiedad">
      [conjunto_propiedades, ...]
   </property>
   [, <property name="nombre_propiedad">
      [conjunto_propiedades, ...]
      </property>
   ] ...

</class>

conjunto_propiedades:
<paramSet name="nombre_conjunto">
   <param name="nombre_parámetro">valor</param>
   [, <param name="nombre_parámetro">valor</param> ] ...
</paramSet>
```

Figura 1. Sintaxis del documento de definición de las clases de entidad (DDCE)

Existen varios tipos comunes de clase, como por ejemplo, *room*, *device*, *person*. Toda clase ha de heredar de uno de estos tipos o de una clase que haya heredado de ellos. La definición de un tipo común contiene las características básicas que pueden compartir todas las clases del mismo tipo. La Figura 2 muestra la definición de la clase de tipo *device* que contiene la propiedad *status*. De este modo, todo dispositivo tendrá una propiedad *status* que determinará si se encuentra encendido o apagado.

```
<classes>
   <class name="device" extends="root">
      <property name="status"/>
   </class>
</classes>
```

Figura 2. Definición de la clase device

Los tipos comunes de clases vienen dados por el sistema y no pueden ser modificados por los usuarios. Los usuarios pueden definir nuevas clases basándose en estos tipos. En la Figura 3 se muestra un ejemplo de una clase luz regulable de tipo dispositivo. La clase hereda la propiedad *status* de la clase *device* y añade una nueva propiedad de tipo *value* (que corresponde con al intensidad de la luz). Por lo tanto, la clase contendrá dos propiedades (*status* y *value*).

```
<classes>
   <class name="dimmerlight" extends="device">
      <property name="value"/>
   </class>
</classes>
```

Figura 3. Ejemplo de un DDCE con la definición de una clase de entidad

3.2.1.1 Parametrización de las clases del entorno

Sobre las clases definidas, se pueden establecer nuevas propiedades y parámetros que añadan nuevas características a la entidad. A su vez, también se pueden dar nuevos valores a propiedades y parámetros definidos anteriormente.

Estas nuevas características se pueden especificar en múltiples documentos diferentes, de forma concurrente. Para ello se emplean los documentos de particularización de las clases de entidad (DPCE). De este modo, varias personas especializadas en aspectos diferentes pueden añadir información de diversa índole a la definición final de la instancia de entidad. Cada una de estas descripciones concurrentes se irá añadiendo a la especificación de la clase establecida en el DDCE (ver apartado anterior), hasta conformar el conjunto completo de propiedades y parámetros que la definen. Estas nuevas propiedades y parámetros se pueden corresponder con la definición de una nueva interfaz, de características comunes, de comunicación con una capa física determinada, etc.

En la Figura 4 se muestra un ejemplo real de cómo se añaden a la definición de las clases *device* y *dimmerlight* los parámetros necesarios para establecer la interfaz Web de todas las instancias que se creen a partir de una clase de tipo luz regulable. Una descripción del proceso de definición y creación de la interfaz Web se puede encontrar en el apartado 3.7.

```
<classes>

   <class name="device">
            <property name="status">
                  <paramSet name="jeoffrey">
                        <param name="type">switch</param>
                        <param name="text_off">Encender</param>
                        <param name="cmd_off">1</param>
                        <param name="text_on">Apagar</param>
                        <param name="cmd_on">0</param>
                        <param name="color_on">0x00FF00</param>
                  </paramSet>
            </property>
   </class>

   <class name="dimmerlight">
            <property name="value">
                  <paramSet name="jeoffrey">
                        <param name="type">slider</param>
                        <param name="lo">0</param>
                        <param name="hi">64</param>
                        <param name="unit">4</param>
                  </paramSet>
            </property>
   </class>

</classes>
```

Figura 4. DPCE que incorpora a device y dimmerlight los parámetros necesarios para crear la interfaz Web

3.2.2 Definición de las entidades de un entorno

Una vez que se tienen definidos los tipos de entidad, se pueden especificar las entidades que se encuentran presentes en el entorno que se está definiendo. Para ello se utilizan los documentos de descripción de las entidades del entorno (DDEE) que especifican las instancias de las clases de entidad que lo componen. Cada instancia hereda las propiedades y parámetros comunes definidos en la clase.

La sintaxis de la definición de una instancia en el DDEE se muestra en la Figura 5. El nombre de la instancia de entidad ha de ser único en todo el entorno y el tipo debe corresponder con alguna de las clases de entidad definidos en los DDCE (ver apartado 3.2.1).

```
<instances>
   definición_instancia
   [, definición_instancia ] ...
</instances>

definición_instancia:
<entity name="nombre" type="clase_entidad"/>
[, <entity name="nombre" type="clase_entidad"/> ] ...
```

Figura 5. Sintaxis de la definición de una instancia de entidad en un DDEE

Por ejemplo, basándonos en los ejemplos del apartado anterior se podría definir una entidad denominada *lampv1* que heredara las propiedades y parámetros de las entidades de tipo luz regulable, tal y como muestra la Figura 6.

```
<instances>
      <entity name="lampv1" type="dimmerlight"/>
</instances>
```

Figura 6. Definición de una entidad a partir de su clase en un DDEE

La instancia *lampv1* poseerá el atributo *status* heredado de *device* (ver Figura 2) y el atributo *value* heredado de *dimmerlight* (ver Figura 3). Cada uno de estos atributos tendrá, además, un conjunto de parámetros de tipo *jeoffrey* (ver Figura 4) que permite establecer automáticamente la interfaz Web de esta instancia de entidad.

3.2.2.1 Parametrización de las entidades en el entorno

De la misma manera que se pueden añadir o modificar las propiedades y parámetros de las clases se puede hacer lo mismo con las instancias. De este modo la definición de la instancia se puede adaptar a las características particulares del entorno en el que ha sido definida. Nuevamente, estas propiedades y parámetros se pueden especificar en múltiples documentos diferentes, denominados documentos de particularización de las entidades del entorno (DPEE), especializándose por el tipo de información que contienen. Cada una de estas descripciones se añade a la especificación de la entidad establecida por los DDEE, hasta conformar el conjunto completo de propiedades y parámetros que la definen.

La definición de nuevos parámetros y propiedades para una entidad sólo se puede hacer sobre instancias de entidades definidas y su sintaxis es muy similar la de la definición de clases de entidad establecida en los DPCE (ver Figura 4).

Por ejemplo, en la Figura 7 se muestra cómo se añaden nuevos parámetros al conjunto *jeoffrey* que definen características concretas de la interfaz Web para la entidad *lampv1* (en este caso su localización en el mapa de la interfaz, ver más detalles sobre la creación de la interfaz Web en el apartado 3.7).

```
<instances>

   <entity name="lampv1">
      <paramSet name="jeoffrey">
         <param name="image">lampv.gif</param>
         <param name="x">650</param>
         <param name="y">130</param>
      </paramSet>
   </entity>

</instances>
```

Figura 7. Incorporación del conjunto de parámetros a la instancia de entidad *lampv1* en un DPEE

De este modo, teniendo en cuenta la definición de los tipos de clase *device* (ver Figura 2), de la clase *dimmerlight* definida por el DDCE de la Figura 3, de los nuevos parámetros de clase para la interfaz Web definidos por el DPCE en la Figura 4, de la definición de la instancia de entidad *lampv1* (ver DDEE de la Figura 6) y de los nuevos parámetros de la interfaz Web definidos para la instancia (ver DPEE de la Figura 7), la entidad estaría formada en su totalidad por los conjuntos de parámetros y propiedades representados en la Figura 8.

```
<entity name="lampv1" type="dimmerlight">
   <paramSet name="jeoffrey">
      <param name="image">lampv.gif</param>
      <param name="x">650</param>
      <param name="y">130</param>
   </paramSet>
   <property name="status">
      <paramSet name="jeoffrey">
         <param name="type">switch</param>
         <param name="text_off">Encender</param>
         <param name="cmd_off">1</param>
         <param name="text_on">Apagar</param>
         <param name="cmd_on">0</param>
         <param name="color_on">0x00FF00</param>
      </paramSet>
   </property>
   <property name="value">
      <paramSet name="jeoffrey">
         <param name="type">slider</param>
         <param name="lo">0</param>
         <param name="hi">64</param>
         <param name="unit">4</param>
      </paramSet>
   </property>
</entity>
```

Figura 8. Disposición final de la entidad lampv1

Todos los conjuntos de parámetros en este ejemplo se refieren a la creación de la interfaz Web. El primero está asociado directamente a la entidad. El segundo se asocia a la propiedad *status* mientras que el tercero lo hace a la propiedad *value*. Una explicación detallada del significado de estos parámetros se realiza en el apartado 3.7

3.2.3 Relaciones entre las entidades del entorno

Además de definir y parametrizar las entidades que forman el entorno se pueden determinar relaciones entre ellas, que establezcan diversos grados de unión.

Por ejemplo, una persona podría relacionarse con un grupo de trabajo específico (lo que determinaría su pertenencia a un grupo u otro) o un sintonizador de radio podría relacionarse con unos altavoces u otros (lo que provocaría que el sonido se emitiera por aquellos con los que está relacionado).

Un caso de espacial interés es de las relaciones entre habitaciones o espacios (lo que establece que un espacio se encuentra dentro de otro, por ejemplo una habitación dentro de una planta) y el de las entidades con habitaciones (que especifica las entidades o elementos que se hayan dentro de la habitación). De este modo se puede crear la representación de un entorno a partir de las habitaciones o espacios que lo componen, su distribución y de las entidades presentes en cada uno de ellos.

El entorno puede componerse de tantas habitaciones o espacios como sea necesario y éstos se pueden encontrar jerarquizados o distribuidos de la misma forma que lo hagan bajo una consideración física o conceptual.

La definición de estas relaciones se realiza en el DPEE de forma similar al modo en que se establecían nuevas propiedades y parámetros de las instancias de entidades (ver apartado anterior). La Figura 9 muestra la sintaxis utilizada para establecer una nueva relación entre dos entidades.

```
<instances>

   <entity name="nombre_entidad_principal">
      <relation name="tipo_relación"
       destination="nombre_entidad_relacionada"/>
   </entity>

</instances>
```

Figura 9. Sintaxis para el establecimiento de relaciones entre entidades en un DPEE

Por ejemplo, en la Figura 10 se representa una entidad denominada *lab_B403* de tipo *room* que se relaciona con la entidad *lampv1* definida en el apartado anterior. Esto implicaría que la entidad *lampv1* se halla dentro de la habitación *lab_B403*.

```
<instances>
   <entity name="lab_B403">
      <relation name="has-resource" destination="lampv1"/>
   </entity>
</instances>
```

Figura 10. DPEE con la definición de la pertenencia de una entidad a una habitación

3.2.4 Documento de descripción del entorno

Toda la información XML especificada en los DDCE (ver apartado 3.2.1), DPCE (ver apartado 3.2.1.1), DDEE (ver apartado 3.2.2) y DPEE (ver apartado 3.2.2.1 y apartado 3.2.3) se une en un único Documento de Descripción del Entorno (DDE). Cualquier persona que quiera conocer cómo está compuesto el entorno sólo tiene que editar el documento y ver sus espacios, entidades, propiedades, parámetros y relaciones.

```
<instances>

   <entity name="lab_B403" type="room">
      <relation name="has-resource" destination="lampv1"/>
   </entity>

   <entity name="lampv1" type="dimmerlight">
      <paramSet name="jeoffrey">
         <param name="image">lampv.gif</param>
         <param name="x">650</param>
         <param name="y">130</param>
      </paramSet>
      <property name="status">
         <paramSet name="jeoffrey">
            <param name="type">switch</param>
            <param name="text_off">Encender</param>
            <param name="cmd_off">1</param>
            <param name="text_on">Apagar</param>
            <param name="cmd_on">0</param>
            <param name="color_on">0x00FF00</param>
         </paramSet>
      </property>
      <property name="value">
         <paramSet name="jeoffrey">
            <param name="type">slider</param>
            <param name="lo">0</param>
            <param name="hi">64</param>
            <param name="unit">4</param>
         </paramSet>
      </property>
   </entity>

</instances>
```

Figura 11. Documento final de descripción del entorno (DDE)

En la Figura 11 se muestra el DDE que describiría los ejemplos vistos desde la Figura 2 a la Figura 10:

Este documento describe que el entorno se compone únicamente de dos instancias. La primera se trata de la habitación *lab_B403*. Mediante la etiqueta *related* se especifica que esta habitación contiene una entidad (*lampv1*) de tipo luz regulable, lo que significa que la habitación contiene una lámpara regulable. A continuación, se declaran las propiedades y parámetros de la entidad *lampv1*.

La información de este documento es empleada por la capa *middleware*, que actúa como una capa de interacción entre el entorno físico, las interfaces de usuario y las aplicaciones.

3.3 La capa middleware

La implementación del *middleware* se basa en una estructura de datos global denominada pizarra [Engelmore, R. and Morgan, T. 1988]. Esta pizarra es un modelo del entorno, donde se encuentra toda la información relativa al mismo. La pizarra proporciona un mecanismo de comunicación asíncrono. Aplicaciones o dispositivos físicos envían información sobre el entorno a la pizarra. Otras aplicaciones pueden suscribirse a estos cambios o extraerlos directamente mediante una consulta.

La información sobre el entorno que se almacena en la pizarra puede verse como una estructura en dos capas. Por un lado una capa de entidades que almacena información sobre cada entidad particular. Por otro lado una capa de relaciones que posee información sobre las relaciones entre las entidades.

La capa de relaciones es un grafo donde cada nodo es una entidad. Cada nodo representa información relevante del entorno, que puede ser un dispositivo físico, una aplicación software, un ocupante o un concepto abstracto. Los arcos entre los nodos denotan algún tipo de relación (de composición, agregación, asociación, etc.). Por ejemplo, la localización de una persona se modela mediante un arco entre la persona y la habitación en la que se encuentra.

En la capa de entidades cada entidad tiene una colección de propiedades y parámetros, tal y como venían definidas en el DDE formado por los documentos XML de descripción del entorno (ver apartado 3.2). La composición de cada entidad del entorno se refleja en la pizarra mediante una estructura en árbol. La raíz de este árbol es un nodo del árbol de relaciones descrito previamente y sus hijos serán el conjunto de propiedades y parámetros. Los valores de las propiedades son nodos hoja que almacenan información de tipo real, entero o cadena de caracteres. Los cambios en los valores de las propiedades que representan variables físicas se reflejan en el mundo físico y viceversa. De este modo, cuando una aplicación necesita obtener o modificar el estado físico de un dispositivo sólo tiene que acceder al nodo adecuado de este grafo y obtener o cambiar su valor.

Combinando estas dos capas la estructura de pizarra resultante se puede ver como un grafo de entidades, donde cada entidad es un árbol de propiedades y parámetros. La Figura 12 representa parte del grafo de una pizarra. La estructura contiene cinco nodos que simbolizan entidades (representados con un círculo continuo y el fondo blanco), cuatro nodos de propiedades (en un círculo continuo y fondo sombreado) y dos nodos de conjunto de propiedades (representados por un círculo discontinuo y fondo blanco) con una propiedad cada uno (en un círculo discontinuo y fondo sombreado). Las flechas representan relaciones bidireccionales y los rectángulos contienen los valores.

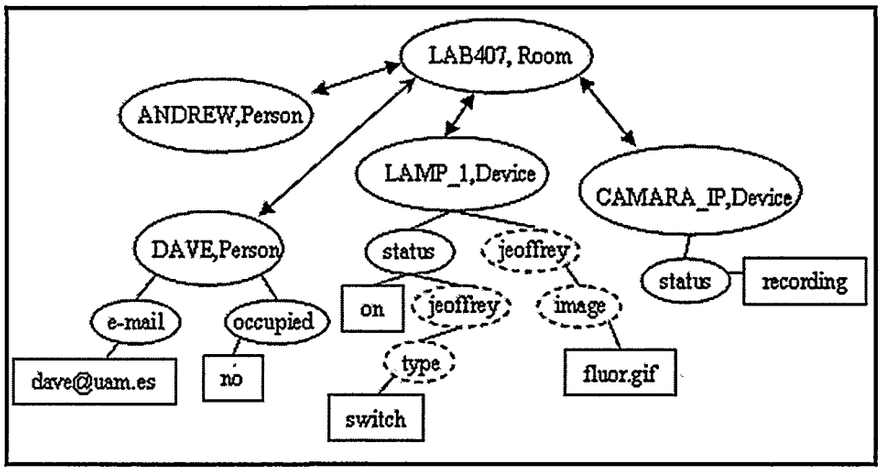


Figura 12. Entidades y sus relaciones en la pizarra

3.4 Interacción con la pizarra

Las aplicaciones no interaccionan directamente con las entidades físicas del entorno o entre sí, sino que sólo tienen acceso a la información del *middleware*. De este modo, los detalles de implementación de una entidad quedan ocultos a las aplicaciones y éstas utilizan las mismas reglas de comunicación estándar para todas las entidades del entorno, independientemente de su naturaleza. A su vez, las entidades físicas o agentes software pueden ser sustituidos por otros que presenten la misma funcionalidad sin necesidad de modificar la información de la pizarra ni las interfaces.

El *middleware* proporciona un conjunto de operaciones que permite extraer información de la pizarra, realizar cambios sobre los valores de las propiedades, añadir o eliminar entidades y relaciones y suscribirse o cancelar la suscripción a eventos de la pizarra. Para operaciones como extraer o cambiar información de la pizarra y añadir o eliminar entidades y relaciones las aplicaciones utilizan mensajes basados en XML que se envían empleando el protocolo HTTP. La Figura 13 muestra el esquema de interacción con el mundo físico de dos aplicaciones a través de la pizarra.

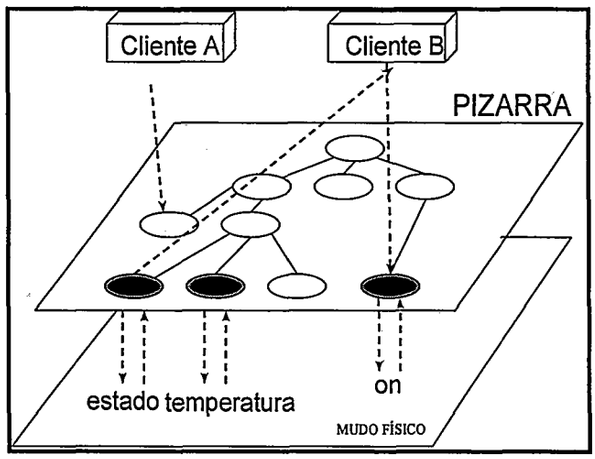


Figura 13. Interacción con la pizarra

Los accesos y consultas se pueden realizar desde los nodos superiores (obteniéndose una representación del árbol consultado) hasta los nodos inferiores (de los que se obtienen valores o características concretos). En todo caso la respuesta también se recibe en forma XML. La Figura 14 muestra un ejemplo de la respuesta obtenida sobre la entidad *lampv1* al consultar el valor de su propiedad *value*.

```
<GetResponse>
   <entity name="lampv1">
      <property name="value">28</property>
   </entity>
</GetResponse>
```

Figura 14. Respuesta obtenida por una aplicación de la pizarra

3.5 Composición del entorno

Utilizando las ideas plasmadas hasta el momento se ha desarrollado un entorno inteligente dentro de un laboratorio habilitado como una sala de estar y reuniones (ver Figura 15). En el entorno se ha colocado un conjunto de dispositivos que "enriquecen" las capacidades del entorno y permiten el desarrollo de aplicaciones de alto nivel para la interacción con el mismo.

Para mostrar, reproducir y acceder a información multimedia se dispone de:

- Un televisor (aunque el control y la interacción con el aparato todavía están en desarrollo).
- Dos altavoces de alta fidelidad que permiten reproducir sonido proveniente de cualquier fuente del entorno.

- Una radio IP (Internet Protocol) con la que se puede seleccionar la emisora que se desea escuchar y modificar el volumen. El sonido se reproducirá por los altavoces que estén relacionados con su entidad radio correspondiente.
- Un reproductor de discos compactos con las características comunes de estos dispositivos: repetir una canción, reproducción aleatoria, etc.
- Unas pantallas planas que muestran información personalizada dependiendo de los ocupantes del entorno.
- Unas cámaras IP que realizarán el reconocimiento de las personas que acceden al entorno (esta aplicación todavía se encuentra en desarrollo) y permiten visualizarlo de forma remota.

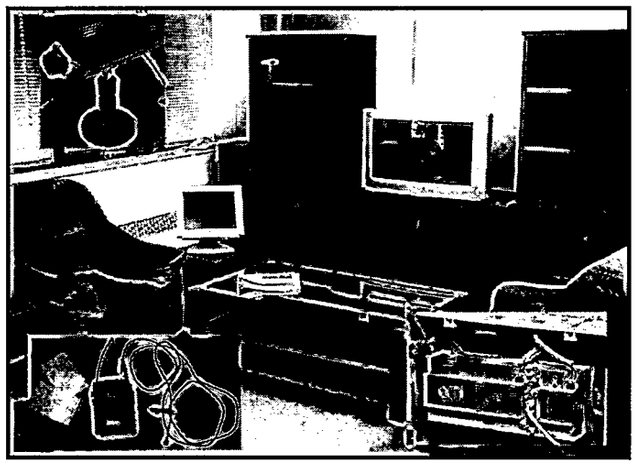


Figura 15. Imagen del entorno inteligente desarrollado

Para el proceso de interacción con voz el entorno posee:

- Un micrófono que se utiliza para registrar la señal de voz.
- Los altavoces anteriormente descritos que también se emplean como medio para reproducir la voz sintetizada.

Dentro de los dispositivos de la red del hogar se encuentran disponibles:

- Una cerradura electrónica que permite controlar el acceso a la puerta principal del entorno.
- Un detector electromagnético que permite conocer si la puerta se encuentra abierta o cerrada.
- Unas tarjetas de proximidad asociadas a cada persona con acceso al entorno. Estas tarjetas permiten accionar la cerradura electrónica y controlar quiénes se encuentran dentro del entorno y en qué momento acceden a él y lo abandonan.

- Un conjunto de relés que se emplea para controlar cinco luces del entorno. La primera de las luces corresponde con los fluorescentes situados en el techo. Las otras cuatro se dividen en dos lámparas de pie. Cada una de las lámparas posee un foco y una luz superior regulable en intensidad.
- Interruptores EIB que permiten el control manual de las luces mencionadas en el punto anterior.

3.6 Aplicaciones desarrolladas en el entorno

Sobre las características del entorno físico desarrollado, y basándose en las funcionalidades que aporta la capa de *middleware,* se ha creado una serie de aplicaciones que "enriquecen" las capacidades del entorno, hacen uso del contexto almacenado en la pizarra y permiten una interacción multimodal con los elementos que lo componen.

Una primera aplicación permite llevar la cuenta de qué personas se encuentran en cada momento dentro del entorno. La aplicación realiza a su vez un saludo oral personalizado cada vez que accede al entorno alguien a quién es capaz de reconocer y una despedida cuando lo abandona. El reconocimiento de la identidad de las personas se realiza por dos vías. La primera se basa en las tarjetas de proximidad personalizadas que dan acceso al entorno. Un receptor situado a la entrada del mismo permite conocer quién entra a él o lo abandona. Una nueva vía que continúa en fase de desarrollo se realiza mediante el reconocimiento de caras. Una cámara situada a la entrada del laboratorio toma una imagen de aquel que accede al entorno y realiza un reconocimiento facial basándose en la base de datos de usuarios del entorno almacenada en la aplicación.

Otra aplicación realiza un control de la seguridad del entorno. Por un lado, basándose en el detector del estado de la puerta, registra el tiempo que ésta permanece abierta de forma continuada. Llegado a un umbral determinado, la aplicación emite un mensaje oral informando que se ha dejado la puerta abierta y se notifica enviando un correo electrónico a los gestores del entorno. Por otro lado se analizan patrones poco habituales de acceso al entorno. Cualquier entrada a partir de una hora específica se considera que puede ser producida por un intruso. Si tras avisar de forma oral de esta anomalía no se produce ninguna reacción se emite una alarma acústica y se envía un mensaje de notificación a los gestores del entorno.

El entorno desarrollado también se emplea como sala de reuniones de grupos de investigación. Una aplicación comprueba cuántos miembros de un mismo grupo se encuentran en el entorno. A partir de alcanzar un cierto umbral se estima que se puede estar realizando una reunión de grupo y se notifica al resto de miembros por correo electrónico. En la actualidad, la aplicación trabaja con dos grupos diferentes.

Una aplicación de información personalizada muestra información específica dependiendo de qué usuarios se encuentren presentes en el entorno. Cada usuario tiene un conjunto de imágenes (pueden ser cuadros, información, etc.) con las que está

relacionado. Para esto basta definir las relaciones entre las personas y las imágenes pertinentes en cada momento. En una de las pantallas planas se muestra continuamente un carrusel de imágenes. Cuando un usuario entra dentro del entorno sus imágenes se añaden automáticamente a las que se muestran en el carrusel. Cuando lo abandona, sus imágenes se eliminan del carrusel y sólo se siguen mostrando las del resto de los miembros presentes en el entorno. De este modo la información que se presenta en las pantallas se adapta automáticamente a los miembros presentes en el entorno.

Otra aplicación gestiona la energía utilizada en el entorno. Si un usuario accede al mismo, y previamente no había nadie presente, la aplicación enciende la luz principal evitando que la persona que acaba de acceder tenga que realizar esta tarea. Por el contrario, si todos los usuarios abandonan el entorno y la luz se queda encendida la aplicación se ocupa de apagarla para evitar el consumo innecesario de energía.

Por último, además de con los interruptores y los métodos de interacción clásicos, los elementos del entorno se pueden controlar mediante dos interfaces diferentes. La primera es una interfaz Web que se describe brevemente en la siguiente sección. La segunda es una interfaz basada en diálogos orales que se describe en detalle en la parte restante de este trabajo.

3.7 La interfaz de control Web

Una de las posibles formas de interacción con el entorno se realiza mediante la interfaz Web. Esta interfaz proporciona una visión parcial del entorno, mostrando aquellos elementos que pueden ser controlados. La interfaz se crea de forma automática, basándose en el Documento de Descripción del Entorno (DDE, ver apartado 3.3). Cabe destacar que a partir de este documento también se crea la capa *middleware* (ver apartado 3.3) y la interfaz de diálogos orales (ver más adelante apartados 5.1 y 5.2). Para la adaptación a la interfaz Web, la información referente a las entidades incluye nueva información específica para la misma.

La interfaz se compone de tres partes estructuradas de forma jerárquica:

- En el nivel superior se encuentra una lista que contiene las habitaciones del entorno. Cuando el usuario selecciona una habitación, aparece la ventana que corresponde con la misma.
- Esta nueva ventana muestra un mapa de la habitación, incluyendo la localización de los muebles y las entidades. El mapa se crea de la composición de una imagen de fondo con las imágenes que representan a los dispositivos. Cada vez que se carga la interfaz el mapa se genera dinámicamente utilizando la información de la pizarra.
- Finalmente, paneles de control específicos se muestran cuando se selecciona una entidad, lo que permite interactuar con ésta.

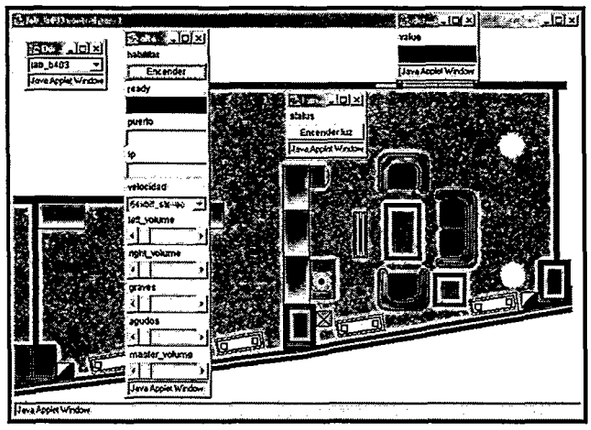


Figura 16. Vista de la interfaz Web

La Figura 16 muestra un ejemplo de una interfaz creada a partir de la información de la pizarra. La ventana de la izquierda corresponde con la lista de habitaciones. La ventana de fondo corresponde con el mapa de una de esas habitaciones. Las otras tres ventanas pertenecen a paneles de control específicos que permiten interactuar con entidades de la habitación. La interfaz Web empieza extrayendo de la pizarra qué habitaciones hay disponibles en el entorno. Luego consulta qué entidades están relacionadas con cada habitación (esto es, qué entidades hay en cada habitación). Por último, obtiene los parámetros y propiedades de cada una de estas entidades.

La definición XML de cada una de las clases e instancias de entidad establecidas en los DDCE y DDEE se puede ampliar con varios parámetros referidos a la interfaz Web que permiten su creación de forma automática (ver apartados 3.2.1.1 y 3.2.2.1). La Figura 17 muestra un DPCE donde se detalla la información que se añade a la clase *lampv1*, tal y como se mostró en la Figura 4.

En primer lugar se añade un conjunto de parámetros a la clase *device*. Estos estarán asociados a la propiedad *status* y se refieren al tipo de control que se ha de mostrar (en este caso un *switch*), el texto que desplegar en el botón cuando la entidad esté apagada (*Encender luz*), qué comando enviar al actuar sobre él al estar apagada (*1*), el texto que hay que desplegar en el botón cuando la luz esté encendida (*Apagar luz*), qué comando enviar al actuar sobre él cuando esté encendida (*0*) y qué color ha de tener al estar encendida (0x00FF00). De este modo, todas las entidades de tipo dispositivo poseerán de forma automática un control para la interfaz Web que permita encenderlo y apagarlo.

A continuación, dado que la clase *dimmerlight*, además del atributo heredado *status* posee un atributo *value*, se añade el conjunto de parámetros necesarios para crear el control necesario de este atributo. Este se asocia al atributo *value* y corresponde con el tipo de control que mostrar (*slider*), el valor en su estado mínimo (*lo*) y máximo (*hi*) y el incremento o decremento que se aplica cada vez que se acciona el control (*unit*). Así,

todas las entidades de tipo *dimmerligh* además de poseer un control para apagar y encender la luz (heredado de *device*) poseerán un control para regularla.

```
<classes>

   <class name="device">
            <property name="status">
                  <paramSet name="jeoffrey">
                        <param name="type">switch</param>
                        <param name="text_off">Encender</param>
                        <param name="cmd_off">1</param>
                        <param name="text_on">Apagar</param>
                        <param name="cmd_on">0</param>
                        <param name="color_on">0x00FF00</param>
                  </paramSet>
            </property>
   </class>

   <class name="dimmerlight">
            <property name="value">
                  <paramSet name="jeoffrey">
                        <param name="type">slider</param>
                        <param name="lo">0</param>
                        <param name="hi">64</param>
                        <param name="unit">4</param>
                  </paramSet>
            </property>
   </class>

</classes>
```

Figura 17. DPCE con la información de la interfaz web de las clases device y lampv1

Una vez establecidos los controles que mostrarán todas las entidades de tipo *device* y *dimmerlight* se pueden determinar parámetros específicos para instancias de esas entidades. La Figura 18 muestra un DPEE donde, además de los parámetros que se definieron en la Figura 7 para una entidad de tipo *dimmerlight*, se representan los que se establecerían para la entidad *lab_B403*. En el caso de la entidad *lab_B403* se especifica qué imagen se va a utilizar de fondo como mapa de la habitación (*background*), cuál va a ser su anchura (*width*) y altura (*height*). Para la instancia *lampv1* se determina qué imagen se va a mostrar en la interfaz *(image)* y su localización dentro del mapa de la habitación (*x* e *y*).

Existen cinco tipos de control diferentes para poder interactuar con las entidades del entorno:

- Áreas de texto, que permiten cambiar el valor de una cadena de caracteres.
- Botones, que actúan como conmutadores de propiedades asociadas a características de apagado y encendido.

- Barras deslizantes, que corresponden a propiedades que toman su valor de un intervalo.
- Listas, que definen una lista de posibles valores entre los que el usuario puede elegir uno.
- Alarma, que corresponden con etiquetas de colores que cambian dependiendo del valor de la propiedad.

```
<instances>

   <entity name="lab_B403">
      <paramSet name="jeoffrey">
                  <param name="background">fondo.jpg</param>
                  <param name="width">764</param>
                  <param name="height">513</param>
      </paramSet>
   </entity>

   <entity name="lampv1">
      <paramSet name="jeoffrey">
                  <param name="image">lampv.gif</param>
                  <param name="x">650</param>
                  <param name="y">130</param>
      </paramSet>
   </entity>

</instances>
```

Figura 18. DPEE con los parámetros empleados por la interfaz web de las instancias lab_B403 y dimmerlight

De este modo cuando un usuario pincha sobre una imagen que pertenece a una entidad del entorno la interfaz lee la descripción de sus propiedades de la pizarra y transforma esta información en un panel de control específico para cada entidad. Si existe más de una propiedad el panel de control se forma de la agregación de los controles de cada propiedad.

La interfaz Web, como el resto de aplicaciones en el entorno, utiliza la pizarra (ver apartados 3.3 y 3.4) como un medio para interaccionar con los elementos del espacio físico (por ejemplo para cambiar el volumen de los altavoces, encender una luz, etc.) y para recibir los cambios que ocurren en el entorno. La interfaz está suscrita a todos los eventos. Los cambios en el estado físico de una entidad se reflejan automáticamente en la pizarra por lo que se muestran inmediatamente en la interfaz. Por ejemplo si una propiedad tiene asociado un control de alarma, la pizarra notificará automáticamente a la interfaz cualquier cambio en el valor de la propiedad, por lo que la interfaz podrá modificar el color del control que tiene asociado.

La interacción con el entorno por medio de la interfaz Web es concurrente con la interacción que se puede realizar mediante la interfaz de diálogos orales (ver más

adelante el apartado 5.3). Ambas interfaces se comunican en todo momento con la pizarra, que sirve como receptor y distribuidor de toda la información referida al entorno. Este hecho permite que coexistan estas dos, u otras posibles, interfaces en un mismo entorno inteligente.

3.8 Estudio del entorno propuesto

En el presente capítulo se han definido algunas de las características que presenta el entorno inteligente que se ha propuesto y desarrollado. A continuación, revisaremos esas características y estableceremos analogías y diferencias con las presentadas en el apartado 2.2 y con los entornos inteligentes descritos en el apartado 2.4.

El entorno inteligente propuesto emplea una red domótica para el control de dispositivos del entorno y una red Ethernet para el control del flujo de información multimedia. Tal y como considera necesario [Pentland, A. 1996] el entorno posee cámaras y micrófonos que transmiten la información a través de estas redes.

Las entidades que forman parte del entorno se definen en un documento XML que sirve de base para la creación automática de uno de sus elementos fundamentales, la pizarra, que sirve como un modelo del mundo. Las entidades se pueden añadir o eliminar del entorno de forma dinámica. Esta incorporación automática y dinámica de dispositivos al entorno y el empleo de un modelo del entorno coincide con las ideas expuestas por [Shafer, S.A.N. 1999].

La pizarra actúa como un *middleware* centralizado que encapsula la interacción de las aplicaciones e interfaces de alto nivel con los elementos físicos del entorno, como se especifica en [Johanson, B. et al. 2002]. Además, sirve como un modelo del mundo que proporciona información de contexto donde cualquier aplicación puede acudir para conocer el estado actual del entorno. Este tipo de *middleware* también se encuentra presente en el proyecto EasyLiving (ver apartado 2.4.5).

Basándose en la información de contexto almacenada, el sistema es capaz de determinar cuál es el estado del entorno, quién se encuentra dentro del mismo y, en algunos casos, qué están haciendo sus ocupantes, tal y como propone [Pentland, A. 1996].

La arquitectura centralizada se opone a la desarrollada por HAL (ver apartado 2.4.1), donde se utilizaba un sistema de agentes distribuidos. La solución centralizada se considera la más óptima como método para almacenar y acceder a la información de contexto, así como para desarrollar de forma más sencilla aplicaciones que interaccionen con el entorno. Un punto en común con HAL radica en el uso de elementos de alto nivel, como por ejemplo de cámaras en el reconocimiento de caras, para obtener la información en el entorno. Sin embargo, en el entorno inteligente propuesto no se abandona la idea de poder utilizar sensores de bajo nivel como un método adecuado para la obtención del contexto, tal y como se realiza en House_n (ver apartado 2.4.4).

Por encima de esta capa se han desarrollado un conjunto de aplicaciones que utilizan la información de contexto almacenada en la pizarra para realizar acciones dentro del entorno. Al igual que ocurre en The Aware Home (ver apartado 2.4.2) las aplicaciones pueden actuar de forma proactiva dependiendo de la situación del entorno. Sin embargo, ese proyecto se centra en mayor medida en un reconocimiento y asistencia a las tareas de los usuarios.

Por último se han creado dos interfaces, una primera interfaz web y una interfaz de diálogos oral (también presente en HAL) que será expuesta en las siguientes secciones. Estas interfaces permiten una interacción ubicua y más natural con el entorno, requisitos expresados como necesarios de forma amplia [Nixon, P. et al. 1999], con la salvedad ya manifestada de The Adaptive House (ver apartado 2.4.3).

El entorno desarrollado se utiliza como sala de reuniones y banco de pruebas para el desarrollo de nuevas aplicaciones. Las aplicaciones desarrolladas se encuentran en continuo funcionamiento y están habilitadas para su uso generalizado, de modo que se puedan ejecutar y probar en ambientes reales. Este ha sido uno de los principios que ha seguido el entorno, al igual que ha ocurrido en otros proyectos (ver iRoom, apartado 2.4.6).

3.9 Interacción con el entorno mediante diálogos orales

Una de las formas prioritarias de comunicación de las personas con el entorno desarrollado se realiza mediante un sistema de habla en lenguaje natural. A su vez, éste responde al usuario utilizando expresiones orales generadas automáticamente. El objetivo es conseguir mantener diálogos robustos en lenguaje natural. Por diálogo robusto se entiende aquel en el que el usuario del sistema percibe que puede emplear expresiones libres y generales (y por tanto no restringidas a un lenguaje predeterminado de comandos) para acceder a las funciones y servicios del entorno.

Para abordar el problema de hacer que los diálogos sean libres y sin restricciones se utilizan tres mecanismos:

- El usuario se encuentra dentro de un entorno restringido que permite poder "intuir" qué tipo de comunicación establecerá con el sistema. Si el habitante del entorno se encuentra en una oficina, no tiene sentido esperar que pregunte por información relativa a elementos del hogar, a no ser que cambie a un modo de interacción con el hogar.
- Aunque el usuario puede establecer múltiples conversaciones con el sistema dentro de un mismo entorno, éste siempre se encuentra dentro de algunos de los diálogos que tiene previstos el sistema. Por lo tanto, si el usuario está llevando a cabo un diálogo cuyo fin es interaccionar con una determinada entidad, el sistema reconoce esta situación, y considera previsible que las próximas expresiones que realice tengan como objetivo terminar de realizar esta tarea.

- El sistema cuenta con una amplia información de contexto que se encuentra almacenada en la pizarra descrita en el apartado 3.3. La información sobre el estado del entorno y los dispositivos físicos que lo componen se almacena en la pizarra para que pueda ser utilizada por los diferentes módulos que componen el sistema, incluido el módulo de lenguaje natural.

3.9.1 Módulos de reconocimiento y síntesis de voz

En esta investigación realizada en del campo de interacción con diálogos orales en entornos inteligentes no se han desarrollado estudios sobre sistemas de decodificación acústica (reconocedores de voz) y de síntesis de voz. Se parte de la base de que el reconocimiento de voz viene dado por un módulo ya construido y a partir de ahí se empieza a realizar el resto de la investigación. Esto es, el trabajo realizado hasta el momento se centra en el módulo de comprensión y generación del habla dentro del entorno.

3.9.1.1 Reconocimiento de voz

El reconocimiento de voz consiste en convertir el lenguaje oral proveniente de un módulo acústico en texto codificado (ver apartado 2.7.2). Los principales pasos en el reconocimiento de voz son:

- Diseño de la gramática. Las gramáticas de reconocimiento definen las palabras que se pueden pronunciar y sus posibles patrones de uso. Los reconocedores crean y activan las gramáticas para saber cuáles son las palabras que pueden percibir como entrada y su posible disposición.
- Procesamiento de la señal. Se analizan las características del espectro de entrada.
- Reconocimiento de fonemas. Se comparan los patrones del espectro con los patrones de los fonemas del lenguaje que se está reconociendo. Un fonema es una unidad básica de sonido de un idioma.
- Reconocimiento de palabras. Se compara la secuencia de fonemas probables con los de las palabras y patrones especificados por las gramáticas activas.
- Generación del resultado. Se proporciona la trascripción de las palabras que ha detectado el reconocedor. El resultado se puede proveer al completar una frase completa o durante el proceso de reconocimiento. A su vez, el resultado puede indicar la mejor de las opciones proporcionadas por el reconocedor o indicar opciones alterativas.

Para el proceso de reconocimiento de la voz se emplea la herramienta comercial ViaVoice de IBM. Se trata de un sistema ampliamente extendido dentro del campo de reconocimiento de voz dirigido fundamentalmente a dictado de texto por usuarios finales. Sin embargo, esta herramienta también se ha utilizado en otros proyectos de similares características, como es el caso de HAL (ver apartado 2.4.1).

Las gramáticas de reconocimiento empleadas se pueden dividir en dos tipos fundamentales: gramáticas de dictado y gramáticas de reglas.

Gramáticas de dictado

Estas gramáticas permiten al usuario expresarse de forma totalmente libre, sin establecer excesivas restricciones. Sin embargo, el coste de esta libertad se presenta en requerir mayores recursos computacionales, una calidad de audio superior y aumentar de forma considerable el número de errores. Las gramáticas de dictado se desarrollan normalmente mediante entrenamiento estadístico con grandes colecciones de texto escrito. El reconocedor ViaVoice utilizado permite el empleo de este tipo de gramáticas.

Gramáticas de reglas

En el reconocimiento del habla basado en reglas es la aplicación la que proporciona al reconocedor las reglas que definen lo que se espera pronuncie el usuario. Así estas reglas restringen el proceso de reconocimiento. Un diseño adecuado de estas reglas, junto con un diseño correcto de la interfaz de usuario, permite a los usuarios tener un grado de libertad de expresión razonable. A su vez, al limitar el rango de lo que puede ser dicho por los usuarios el proceso de reconocimiento será más rápido y preciso (ver apartado 2.8.3). El reconocedor utilizado también permite el empleo de estas gramáticas de reglas, siendo este el modo empleado en el sistema desarrollado.

3.9.1.2 Síntesis de voz

La síntesis de voz (conocida como Text To Speech o TTS) consiste en convertir texto escrito en lenguaje hablado. Los principales pasos para conseguir este resultado son:

- Análisis de la estructura. Se procesa el texto de entrada determinando dónde empiezan y terminan los párrafos, oraciones y otras estructuras. Generalmente se utiliza para tal fin la puntuación y el formato de los datos.
- Preprocesamiento del texto. Se analiza el texto para construcciones especiales del lenguaje. Se necesita un tratamiento especial para el uso de abreviaturas, fechas, acrónimos, etc.
- Conversión de texto a fonemas. Se convierte cada palabra en fonemas.
- Análisis de la prosodia. Procesa la estructura de la oración, sus palabras y fonemas para determinar la prosodia adecuada para la oración. La prosodia incluye el tono, la cadencia, las pausas, énfasis, etc.
- Producción de la salida. Los fonemas y la información de prosodia se utilizan para generar la salida audio de la oración. Este proceso se puede realizar concatenando pedazos de habla grabada o utilizando técnicas basadas en el conocimiento de cómo suenan los fonemas y cómo les afecta la prosodia.

El sistema ViaVoice de IBM posee estas capacidades y permite realizar síntesis de voz en español con unos niveles de entendimiento bastante satisfactorios. Además, esta

herramienta permite ajustar la prosodia lo que permite que el usuario pueda percibir mayor sensación de naturalidad.

3.9.1.3 API de comunicación

La comunicación con el reconocedor y sintetizador se realiza utilizando la Java Speech API. Se trata de una extensión de la plataforma Java que proporciona una interfaz software multiplataforma e independiente del motor de reconocimiento para el desarrollo de aplicaciones de interacciones orales. Su empleo garantiza que se pueda cambiar en cualquier momento el software o hardware de reconocimiento o síntesis de voz sin que esto afecte a la implementación de la aplicación. Alguna de las características que presenta esta interfaz son:

- Es monolingüe, sólo puede utilizar con un único lenguaje especificado.
- Procesa una única señal de entrada de audio.
- Se puede de forma opcional adaptar a las voces de los usuarios.
- Sus gramáticas se pueden adaptar de forma dinámica.
- Permite controlar la prosodia de la síntesis. En el caso de que el sintetizador lo permita se puede regular el volumen, la velocidad del habla y el tono.
- Si el sintetizador presenta varias voces se permite seleccionar con qué voz se desea reproducir el texto.

3.9.2 Consideraciones sobre el reconocimiento de voz establecidas en los sistemas de diálogo diseñados

Los modelos de sistemas de diálogos estudiados y desarrollados se basan en tres premisas relativas al reconocimiento: no se realiza ningún entrenamiento al reconocedor de voz, se emplean gramáticas de reglas para mejorar la tasa reconocimiento y se debe invocar explícitamente al reconocedor para iniciar la interacción con el entorno. La primera de ellas reduce la tasa de reconocimiento pero amplía su posibilidad de uso a cualquier usuario. La segunda limita la libertad del usuario para establecer una interacción con el entorno pero mejora sensiblemente la eficacia del reconocedor. La última evita reconocer oraciones que no van dirigidas a la interfaz de diálogos.

3.9.2.1 Reconocimiento de voz sin entrenamiento

Una de las ideas básicas en el proceso de reconocimiento en los sistemas de diálogos diseñados consiste en que el reconocedor de voz no debería estar entrenado para ningún usuario específico.

Es cierto que en entornos restringidos (como son los del hogar) no resultaría difícil realizar un entrenamiento del reconocedor de voz para que éste se adaptara a cada uno de sus ocupantes. En este caso problema surgiría a la hora de decidir qué usuario es el

que está interactuando con el sistema en cada momento, aunque para esto existen técnicas de reconocimiento del hablante (speaker recognition) o seguimiento del hablante (speaker tracking). Sin embargo, en entornos más abiertos no resulta tan fácil restringir el número de usuarios del entorno, e incluso el sistema puede ser utilizado por nuevos usuarios cada vez.

Dadas estas consideraciones, y teniendo en cuenta que siempre se ha intentado realizar un sistema que pudiera ser aplicado a los campos más generales posibles, todas los estudios y pruebas realizadas con los sistemas de diálogos se basan en reconocedor de voz no entrenado para ningún usuario específico.

3.9.2.2 Empleo de gramáticas de reglas en el reconocimiento

Inicialmente se hicieron pruebas de reconocimiento sobre un modelo basado en la gramática de dictado del reconocedor sin establecer ningún tipo de restricción. Pronto se comprobó que el empleo de este tipo de técnicas resulta inviable dado el nivel de desarrollo actual de la tecnología de reconocimiento, de manera que era necesario utilizar gramáticas de reglas.

Una ventaja del empleo de estas gramáticas consiste en que las posibles interacciones dentro de un entorno son altamente limitadas. Aunque en entornos diferentes las posibilidades de interacción son muy diversas, sólo es necesario considerar aquellas interacciones que tienen sentido dentro del entorno con el que el usuario está interaccionando. Con esto, el usuario sigue manteniendo la sensación de un alto grado de libertad en sus posibles interacciones con el entorno y la tasa de reconocimiento se incrementa de forma considerable.

El formato de gramáticas utilizado es JSGF (Java Speech Grammar Format). Se trata de un tipo de gramáticas regulares independiente de la plataforma y del reconocedor que es compatible con la Java Speech API. En la Figura 19 se muestra un ejemplo de una gramática de reglas sencilla empleando el JSGF.

```
#JSGF V1.0;
// Define el nombre de la gramática
grammar GramaticaSencilla;
// Define las reglas
public <Comando> = [<Urbanidad>] <Accion> <Objeto> (y <Objeto>)*;
<Accion> = abre {abrir} | cierra {cerrar};
<Objeto> = la puerta {puerta} | la ventana {ventana};
<Urbanidad> = por favor;
```

Figura 19. Ejemplo de gramática en formato JSGF

Los nombres de las reglas están representados entre símbolos de menor y mayor. Las palabras que se pueden pronunciar se escriben sin ningún tipo de acompañamiento. La gramática se basa en un regla pública, *<comando>*, (que es la que se puede pronunciar) basada en tres subreglas. Los elementos situados entre corchetes son opcionales. Los paréntesis agrupan partes y el asterisco indica que ese elemento puede ocurrir cero o

más veces. Esta gramática permite a los usuarios pronunciar oraciones del tipo: *abre la puerta* o *por favor cierra la puerta y la ventana*. Por último, las partes representadas entre llaves indican cómo se deben interpretar los elementos reconocidos. Por ejemplo, si el usuario pronuncia la frase *por favor cierra la puerta* el reconocedor sólo devolverá a la aplicación las palabras *cerrar puerta*.

Por lo tanto este tipo de gramática por un lado determina cuáles son las frases que puede recibir el reconocedor de voz y por otro interpreta determinadas secuencias de palabras. Esta secuencia de palabras es la que se envía a la aplicación que invocó el reconocimiento, simplificando el procesamiento posterior de esta información.

3.9.2.3 Respuesta del reconocedor

Como medida de precaución para evitar confusiones con las oraciones de ámbito general que pueda pronunciar un usuario (por ejemplo, si está hablando con otra persona), se consideró necesario que los sistemas de diálogos no siempre *presten atención* a lo que se está diciendo en el entorno.

De este modo el sistema se debe encontrar originalmente en un estado *dormido*, donde no se atiende a lo que se dice. Para activar la interacción con el entorno se debe, tras una breve pausa, pronunciar la palabra *Odisea*. Entonces el sistema *despierta* y responde al usuario para que éste sea consciente de que puede iniciar la interacción con el entorno. A partir de ese momento la interfaz de diálogos considera que todas las oraciones pronunciadas por el usuario tienen como fin realizar alguna acción con el entorno y *permanece atento* a toda oración pronunciada por el usuario.

Si transcurre un número determinado de segundos desde que el sistema se ha despertado sin que se produzca ninguna interacción por parte del usuario, o si transcurre ese tiempo desde la última interacción del sistema o del usuario, la interfaz vuelve al estado de *dormido*, avisando de que se ha producido esta circunstancia. Si el usuario quiere volver a interaccionar con el entorno deberá volver a *despertar* al sistema de diálogos.

Esta medida también se adopta en otros entornos inteligentes que presentan interacción oral con los usuarios, como es el caso de HAL (ver apartado 2.4.1).

4 Primer modelo de interacción

"El retirar no es huir, ni el esperar es cordura, cuando el peligro sobrepuja a la esperanza; y de sabios es guardarse hoy para mañana, y no aventurarse todo en un día"
Sancho – Capítulo XXIII – Primera Parte

Inicialmente se diseñó un primer sistema de diálogos orales para controlar los elementos del entorno. El sistema utilizaba un modelo basado en tareas donde cada diálogo correspondía con una tarea. Un supervisor se encargaba de decidir qué tarea quería realizar el usuario y determinar así qué diálogo se ejecutaba.

El sistema se implementó y se evaluó durante tres días en una feria científica de carácter divulgativa con público diverso. Los resultados de este trabajo han servido para asentar las bases necesarias para la especificación de este tipo de interfaces.

Este primer sistema de diálogos fue finalmente descartado dada su rigidez, que le hacía muy difícil adaptarse a las características dinámicas de los entornos inteligentes. En su lugar se ha desarrollado un sistema de diálogos de mayor envergadura que se amolda en gran medida a estos entornos. Este sistema es objeto de un estudio detallado dentro del capítulo 5.

El sistema que se describe en el presente capítulo estaba fundado en los sistemas basados en plantillas (frame-based systems), comentados en la siguiente sección.

4.1 Sistemas basados en plantillas

En los sistemas basados en plantillas [McTear, M.F. 2002] la interacción entre el usuario y el sistema permite rellenar huecos en una plantilla con el fin de realizar una tarea. Estos sistemas son análogos a un proceso de relleno de un formulario en donde se debe recopilar un conjunto predeterminado de datos. La plantilla lleva la cuenta de los elementos para los cuales el sistema requiere información. El sistema puede establecer qué preguntas puede realizar al usuario en el proceso de recogida de la información necesaria para completar la tarea. Además, otros factores como el contexto ayudan a determinar qué preguntas e información pueden ser requeridas. Estos datos se pueden utilizar en conjunción con guiones de conversación, que establecen los caminos que puede seguir un diálogo.

Los sistemas basados en plantillas no se basan necesariamente en un cuestionario rígido: el usuario puede proporcionar múltiple información a la vez, rellenando diversos huecos de la plantilla. A su vez, el sistema puede adaptar la respuesta a las partes del formulario que queden por rellenar. Así permiten no realizar más preguntas ni realizar más confirmaciones de las que resulten necesarias y utilizar la información que ha proporcionado el usuario aunque el sistema no la hubiera requerido todavía. Estos sistemas también permiten una aproximación a modelos de iniciativa mixta (ver apartado 2.9.2.1). El sistema puede guiar a los usuarios para completar una plantilla o bien éstos pueden iniciar un nuevo diálogo completando datos de una nueva plantilla.

4.2 Composición de los diálogos

El sistema de diálogos desarrollado se compone de un diálogo para cada tarea posible. Por ejemplo, el diálogo de luces controla las lámparas de una habitación. De este modo,

dado que los diálogos estaban orientados a tareas, el diálogo de luces debe controlar todas las luces del entorno.

Estos diálogos se basan en la idea de guiones de conversación y plantillas de tarea:

- Los guiones de conversación se componen de una secuencia de patrones de palabras, con sus respectivas relaciones, y de sus respuestas correspondientes. Las frases pronunciadas por el usuario, y adecuadamente analizadas, se comparan con la colección de patrones de palabras almacenadas en el guión de la conversación. Cuando se produce una coincidencia se ejecuta la parte del guión correspondiente, lo que puede desencadenar la realización de ciertas acciones o de una respuesta al usuario. Para ello también se utilizaba la información de contexto almacenada en la pizarra.
- Las plantillas de tarea definen los parámetros requeridos para completar una determinada tarea. Cuando se produce una coincidencia con un patrón, el guión de conversación correspondiente puede invocar el procesamiento de la frase con respecto a una plantilla de tarea específica hasta que ésta se complete.

El modelo empleado es similar al concepto de *script* de [Schank, R. and Abelson, R. 1977]. Un diálogo se forma por una plantilla con huecos que se deben rellenar. El diálogo guía al usuario a través de su guión hasta que se completa la plantilla.

Los diálogos son independientes entre sí y cada uno tiene su propia gramática asociada. No intercambian información directamente y no son conscientes del estado de los otros diálogos. La forma en que los diálogos pueden intercambiar información con los demás o con otras aplicaciones se realiza colocando la información en la pizarra (ver apartado 3.3).

La plantilla de un diálogo no tiene que rellenarse necesariamente con una única oración. En ese caso el sistema de diálogos mantiene su estado de modo que se pueda completar con futuras oraciones. En el proceso de relleno de la plantilla el diálogo puede modificar o leer la información de la pizarra, de modo que puede informar al resto del mundo de un nuevo estado o puede obtener ayuda para finalizar su tarea actual correctamente.

Cuando se completa un diálogo éste debe realizar las acciones que lleve asociadas. Estas pueden variar desde realizar una acción física en el entorno hasta proporcionar información al usuario. En el caso de tener que modificar el estado del entorno el diálogo realiza la acción accediendo a la pizarra y nunca comunicándose directamente con las entidades del entorno.

4.3 Selección del diálogo

Todos los posibles diálogos del entorno discurren en paralelo, en diferentes hilos de ejecución. Sin embargo sólo uno de estos diálogos puede tomar el control de la situación. De este modo los diálogos tienen que competir para conseguir gestionar cada

interacción. Un supervisor de diálogos se ocupa de dar paso al diálogo más adecuado. Su funcionamiento sigue los siguientes pasos (ver Figura 20):

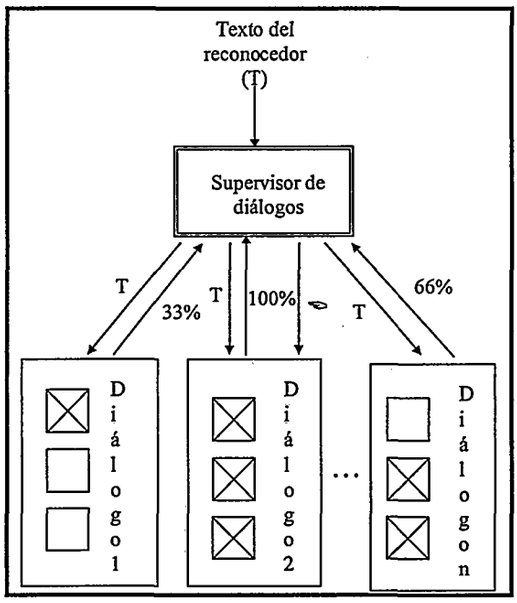


Figura 20. Arquitectura del sistema de diálogos inicial

- El supervisor recibe la oración interpretada por el reconocedor de voz.
- A continuación la envía a todos los diálogos, que se encuentran en estado de espera.
- Cada diálogo recibe la oración y la compara con su plantilla con el fin de descubrir cuántos huecos puede rellenar y cuántos dejaría vacíos.
- Cada diálogo responde al supervisor con el porcentaje de huecos rellenos, teniendo en cuenta aquellos que se rellenaron con oraciones previas y los que se completan con la actual oración.
- El supervisor recibe esta información de todos los diálogos y comprueba cuál tiene un porcentaje mayor.
- Si hubiera múltiples diálogos con el mismo porcentaje el supervisor considerará a aquel que fue seleccionado en interacciones previas, con el fin de continuar con la tarea que se inició previamente.
- Una vez obtenido el resultado el supervisor cede el control a este diálogo, que será el único que podrá continuar el proceso de interacción y realizar alguna tarea.
- El resto de diálogos esperará a recibir una nueva oración para volver a competir.

Dado que el supervisor y cada uno de los diálogos se ejecutan en hilos distintos, una vez que el supervisor ha elegido al diálogo más adecuado el supervisor puede recibir

una nueva oración del reconocedor, independientemente de que el diálogo seleccionado se encuentre en mitad de su proceso. En este caso el supervisor espera hasta que el diálogo acabe y sólo entonces emplea la nueva oración para enviársela a todos los diálogos. Esto se utiliza fundamentalmente para permitir al usuario responder durante la respuesta sintetizada de un diálogo, sin tener que esperar a que esta termine. Aunque al recibir una nueva oración el supervisor no interrumpe el diálogo en curso la oración se almacena para ser utilizada tan pronto como sea posible.

4.4 Descripción de los módulos del sistema

El supervisor del sistema se compone de dos módulos fundamentales que garantizan el correcto intercambio de oraciones y la selección adecuada del diálogo. Son el Módulo de Intercambio de Oraciones (MIO) y el Módulo de Selección del Diálogo (MSD).

4.4.1 El Módulo de Intercambio de Oraciones

El MIO se ocupa de enviar las oraciones a cada uno de los diálogos y, dado que todos los diálogos se procesan de forma concurrente, se asegura de que cada oración es recibida y procesada por cada uno de ellos. Un esquema de la composición del Módulo de Intercambio de Oraciones se muestra en la Figura 21.

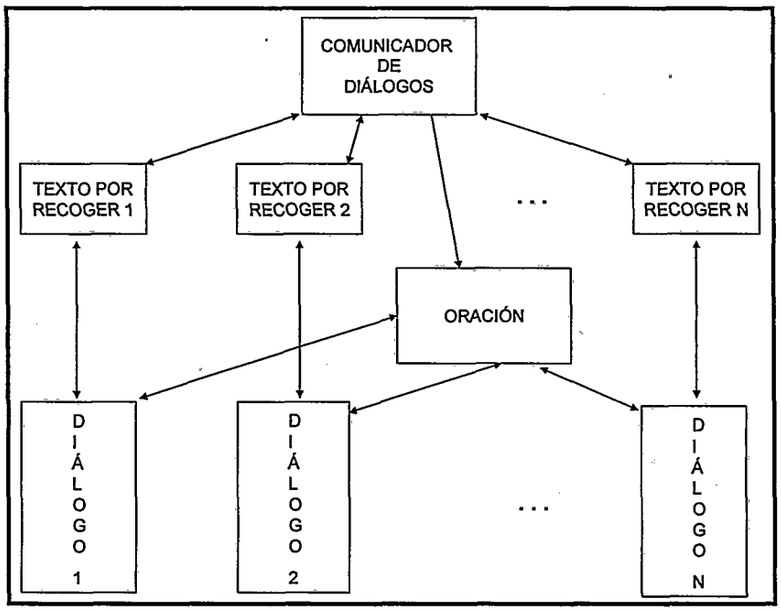


Figura 21. Esquema del Módulo de Intercambio de Oraciones

Por cada diálogo, el MIO crea una unidad de intercambio de información. El valor de esta unidad será *verdadero* o *falso* dependiendo de si el último texto ha sido recogido por el diálogo o no.

El proceso se inicia cuando el *Comunicador de diálogos* recibe una nueva oración. Entonces comprueba el valor de todas las unidades *Texto por recoger*. En el caso de que alguna devuelva el valor *verdadero* (esto es, el diálogo correspondiente no haya todavía recogido la oración anterior), el comunicador esperará a que su valor cambie a *falso*. Una vez que el *Comunicador de diálogos* se ha cerciorado de que todos los diálogos recogieron la última oración (todas las unidades *Texto por recoger* devolvieron falso), coloca la nueva oración en la unidad de *Recogida de oración*. A continuación notifica a los diálogos que existe una nueva oración enviando el mensaje *verdadero* a todas las unidades *Texto por recoger*.

Cada uno de los diálogos comprueba continuamente el estado de su unidad *Texto por recoger*. Si en algún momento su unidad le devuelve el valor *verdadero* acude a la unidad de *Recogida de oración* para obtener la nueva oración. En ese momento envía además el mensaje *falso* a su unidad *Texto por recoger* correspondiente, con el fin de informar de que ya ha recogido la oración correspondiente.

Este diseño permite una absoluta concurrencia entre las diferentes unidades del módulo y los diálogos. Al mismo tiempo que los diálogos procesan la última oración recibida, el *Comunicador de diálogos* puede enviar nuevas oraciones que pueden ser procesadas por diálogos que ya hayan terminado con la anterior.

4.4.2 El Módulo de Selección del Diálogo

Una vez que un diálogo ha procesado la oración recibida, éste responde al MSD con su estimación de prioridad para el procesamiento de la oración. Este valor se toma considerando el número de huecos de la plantilla rellenos y por rellenar (ver apartado 4.3). Nuevamente, el MSD permite la misma concurrencia al utilizar unidades de *Prioridades por recoger* y *Notificaciones por recoger*. El esquema de su composición se muestra en la Figura 22.

Una vez que el diálogo ha procesado la oración recibida y calculado su estimación de prioridad, envía la información al *gestor de prioridades*. Sin embargo, antes comprueba que éste haya recogido la última prioridad que envió. En caso contrario, esperará a que esto se produzca. A su vez, el *gestor de prioridades* comprueba continuamente el estado de las unidades *prioridad por recoger* de cada diálogo. En el caso de recibir una nueva prioridad, la recoge y notifica de este hecho.

Cuando el *gestor de prioridades* ha recogido todas las prioridades de un mismo turno de los diálogos, calcula cuál es el diálogo de mayor prioridad. En caso de recibir prioridades iguales para diferentes diálogos escoge el diálogo que se procesó en la última interacción, con el fin de continuar con la tarea que se está realizando (ver apartado 4.3) o, en su defecto, el primero de ellos.

Esta información se transmite al *Selector de diálogo* que, utilizando un mecanismo concurrente análogo al descrito, informa a cada diálogo si ha sido elegido o no para realizar la tarea que lleve asociada.

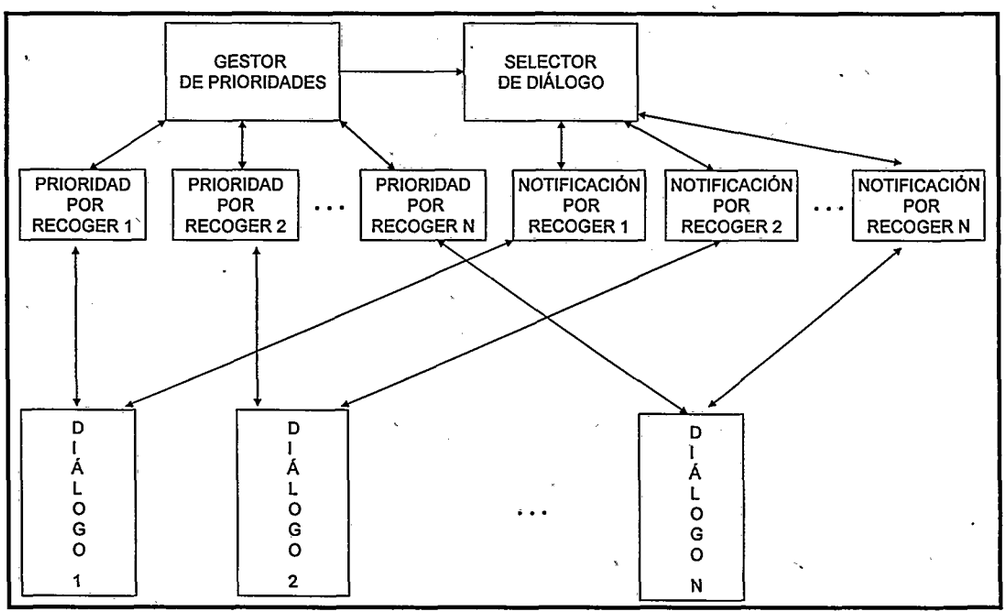


Figura 22. Esquema del Módulo de Selección del Diálogo

4.5 Evaluación inicial

Este sistema fue utilizado durante cuatro días en un evento científico abierto al público en general. La gente pudo utilizar el sistema de diálogos y cambiar el estado físico del laboratorio. Los resultados se podían comprobar desde una webcam conectada con el laboratorio que emitía la imagen en tiempo real.

El sistema fue utilizado por múltiples usuarios, en su mayoría no familiarizados con las tecnologías de la información, de ambos sexos y de un amplio abanico generacional. Estos no recibieron instrucciones previas sobre cómo debían hablar para interaccionar con el sistema y sólo se les dijo que el sistema era capaz de reconocer sus órdenes.

En la mayor parte de los casos la interacción con el sistema fue satisfactoria y se consiguió interactuar con el sistema de forma sencilla. Sin embargo no se realizaron medidas del reconocimiento, interpretación y éxito de las tareas, ni se rellenaron cuestionarios para conocer el grado de satisfacción subjetiva de los usuarios. Estas pruebas fueron utilizadas para obtener un modelo textual de cómo la gente interacciona con las entidades del entorno, añadiendo y considerando las nuevas formas de interacción obtenidas en los futuros desarrollos del sistema.

4.6 Estudio sobre VoiceXML

Uno de los mayores problemas del sistema de diálogos descrito hasta ahora es que la definición e implementación de los diálogos era muy cerrada y centrada a un entorno específico. Dado que los entornos se describen y representan de forma estándar y que

esta descripción se realiza empleando etiquetas XML (ver apartado 3.2) se estudió la posibilidad de estandarización de la definición de los diálogos de interacción con el entorno.

Para ello se consideró el empleo de VoiceXML (Voice eXtensible Markup Language) como herramienta de definición de diálogos. VoiceXML es un lenguaje de etiquetas basado en XML para crear aplicaciones de voz distribuidas que permite emplear síntesis de voz, audio digitalizado, reconocimiento del habla, marcación de teclado telefónico (Dual Tone Multi-Frequency, DTMF), grabación del habla, telefonía y diálogos de iniciativa mixta (ver apartado 2.9.2.1).

VoiceXML proporciona un entorno abierto con una descripción de diálogos y formato de gramáticas estándar. Un documento VoiceXML especifica cada diálogo de interacción que va a ser llevado a cabo por un intérprete. Además configura una máquina conversacional de estados finitos con un cierto grado de iniciativa mixta que permite a los usuarios introducir de forma limitada más de un valor en un estado particular del diálogo.

VoiceXML presenta como similitud con el sistema de diálogo desarrollado el empleo de formularios como el elemento principal de los diálogos [McTear, M.F. 2002]. Un formulario en VoiceXML consta de un campo y unos elementos de control. Un campo recoge la información del usuario utilizando la entrada por voz o DTMF mientras que los elementos de control involucran secuencias de declaraciones empleadas para generar respuestas o realizar procesamientos. El proceso se basa en el algoritmo de interpretación del formulario (Form Interpretation Algorithm) que determina qué elementos visitar en el formulario. Los formularios pueden estar dirigidos o presentar iniciativa mixta. En el primer caso, los elementos se ejecutan una vez en orden secuencial, dando como resultado en un diálogo rígido dirigido por el sistema. En un formulario de iniciativa mixta, combinado con una gramática, permite al usuario introducir todos los elementos requeridos en una sola oración, produciendo un diálogo más flexible.

Para comprobar las posibilidades de empleo de VoiceXML se realizó la integración de un reconocedor de voz con un intérprete VoiceXML y un sintetizador de voz. Este trabajo fue llevado a cabo en el Electrical and Computer Engineering Department de la University of Miami. El reconocedor de voz elegido fue Sphinx [http://cmusphinx.sourceforge.net/html/cmusphinx.php], un motor de reconocimento del habla de código abierto desarrollado en la Carnegie Mellon University. El sintentizador de voz utilizado fue Festival [http://www.cstr.ed.ac.uk/projects/festival/], desarrollado en el Centre for Speech Technology Research de la University of Edinburgh. El intérprete seleccionado fue OpenVXI [http://fife.speech.cs.cmu.edu/openvxi/], también de la Carnegie Mellon University. Se trata de una libería de código abierto que interpreta VoiceXML y con las capacidades de reconocimiento y síntesis simuladas, por lo que resultó necesario integrarlas con los otros sistemas. Este no resulta un proceso trivial y su consecución generó interés en la

creación, junto con otros desarrolladores, de un módulo de código abierto con licencia pública (Lesser General Public License, LGPL) que integrara los tres elementos.

A partir de ese momento se definieron unos diálogos iniciales que permitieran comprobar sus posibilidades de uso dentro del sistema. En ningún momento se integraron estos diálogos con el entorno desarrollado (ver apartado 3.5) por lo que el trabajo se limitó a la realización de simulaciones. Como principal problema para su uso en el entorno propuesto se encontró que VoiceXML presenta un modelo de diálogos excesivamente rígido, basado en gran medida en un modelo de máquina de estados finitos, que no puede adaptarse a la complejidad y el dinamismo en que se basan los entornos inteligentes.

La actual versión de VoiceXML se ha aceptado como un estándar para sistemas de diálogos hablados basados en la Web. Además, en los próximos años, servidores de VoiceXML abiertos pueden reemplazar a plataformas propietarias para sistemas de diálogos orales. Sin embargo, futuras versiones han de incorporar nuevas características que hagan posible el uso de las funcionalidades avanzadas que requiere la dinámica de los diálogos que se desarrollan dentro de un entorno inteligente.

4.7 Herramientas para el desarrollo de sistemas de diálogos orales

Considerando las características dinámicas de los entornos inteligentes, otro requisito necesario para el desarrollo de un sistema de diálogos es la facilidad en el diseño del sistema a partir de la composición del entorno. La definición de los diálogos debe adaptarse fácilmente a entornos muy heterogéneos.

En este sentido se estudió la funcionalidad de las herramientas (toolkits) para el desarrollo de sistemas de diálogos orales. Estas herramientas permiten la construcción de sistemas de diálogos orales incluso a aquellas personas que no estén especializadas en las tecnologías que los componen, como reconocimiento del habla o procesamiento del lenguaje natural.

Existen diversas herramientas disponibles, tanto desde el punto de vista de la investigación (CSLU toolkit) como del comercial (Nuance Developers' Toolkit o SpeechMania). Concretamente se ha prestado especial atención a la CSLU toolkit [http://cslu.cse.ogi.edu/toolkit/], desarrollada en el Center for Spoken Language Understanding (CSLU) del Oregon Graduate Institute os Science and Technology.

La herramienta dispone de una interfaz gráfica para el desarrollo rápido de aplicaciones (RAD) en el campo de las interfaces de diálogo. La ventaja de este RAD se centra en el hecho de que los usuarios quedan aislados de muchos de los procesos complejos que se dan en la construcción de una interfaz de diálogos orales. Para construir un diálogo basta con seleccionar y enlazar objetos gráficos de diálogo en un modelo de diálogo de estados finitos, que puede incluir división de caminos, ciclos, saltos o subdiálogos.

En general, la mayoría de estas herramientas proporciona algún tipo de soporte a la programación visual del sistema [McTear, M.F. 2002]. Además permiten algún tipo de comprensión del lenguaje natural, generalmente con una gramática basada en conceptos o semántica con términos que se corresponden directamente con elementos de dominios específicos. Sin embargo no presentan apoyo a la construcción de gramáticas. Asimismo proporcionan sus propios sistemas de reconocimiento del habla y utilidades para la reutilización de los componentes desarrollados.

A pesar de estas características, este tipo de herramientas tampoco resulta adecuado para el desarrollo de interfaces de diálogos orales para entornos inteligentes. El tipo de diálogos que se desarrolla resulta excesivamente sencillo y orientado a dominios concretos, lo que no permite representar la complejidad y diversidad de los entornos inteligentes y aprovechar la información de contexto que proporcionan. Además, no se adaptan fácilmente a las características dinámicas que estos entornos presentan.

4.8 Limitaciones del modelo

Como se ha mencionado a lo largo de este capítulo, los modelos propuestos no ofrecen una solución satisfactoria ante la complejidad de la construcción de sistemas de diálogos orales para entornos inteligentes.

Los diálogos desarrollados como primer modelo eran completos e independientes y estaban orientados a tareas. Esto ocasionaba que si se añadía o eliminaba una entidad en el entorno, para ser consistente con la nueva situación se debía modificar el diálogo que agrupaba a todas las entidades que realizaban una tarea similar.

Por un lado, esta orientación incrementaba la complejidad de la creación del diálogo cuando se añadían entidades similares. Por otro lado, si se eliminaba una entidad del entorno adaptar el diálogo a la nueva situación podría no resultar una tarea trivial.

Estas limitaciones contrastan con las ideas expuestas en el capítulo 2, donde se especificaba que los entornos inteligentes son espacios altamente dinámicos cuya configuración puede cambiar de forma considerable. Siguiendo esta pauta se considera imprescindible que la interfaz de diálogos orales se pueda crear y adaptar de forma automática a las características del entorno, sin necesidad de la supervisión de un experto cada vez que se produce un cambio.

Considerando estas ideas, se decidió ampliar la funcionalidad del modelo de diálogos para que fuera capaz de contener las funcionalidades que se han mostrado a lo largo de este capítulo.

En primer lugar se debería aprovechar la definición que se realiza en XML del entorno (ver apartado 3.2) para poder definir, a su vez, la interfaz de diálogos orales. Esta definición traería consigo varias ventajas. Por un lado se conseguiría un mecanismo estándar de definición de los diálogos, empleando el lenguaje XML (y que fuera más versátil que VoiceXML dentro del campo de los entornos inteligentes). Por otro, se crearía una herramienta de desarrollo de diálogos que pudiera ser utilizada por

un público general, abstrayéndole del uso de los conocimientos requeridos en la tecnología del habla (similar a los *tookits* existentes, pero aprovechando muchas de las características que proporcionan los entornos estudiados).

En segundo lugar se debería establecer como un aspecto relevante la posibilidad de emplear una definición automática de los diálogos, que se adaptara a las características dinámicas de los entornos. El sistema basado en plantillas desarrollado se considera una herramienta suficiente como base para llevar a cabo los diálogos de interacción con el entorno. Sin embargo, su estructura debía ser modificada para permitir una definición dinámica y automática de los diálogos.

Considerando todas estas ideas se llevó a la modificación del primer modelo propuesto para crear un nuevo sistema que se creara automáticamente a partir de una definición estándar del entorno y que permitiera adaptar las posibles interacciones a cada uno de los entornos sin necesidad de realizar ningún cambio en su estructura. Este nuevo sistema también basó sus ideas en el sistema de plantillas desarrollado aunque modificándolo, mediante una estructura en árbol, para permitir la definición automática de los diálogos.

5 Modelo de diálogos orales automáticos propuesto para la interacción con el entorno inteligente

"Cada uno es hijo de sus obras"
Don Quijote – Capítulo IV – Primera Parte
Sancho – Capítulo XLVII – Primera Parte

"La senda de la virtud es muy estrecha, y el camino del vicio ancho y espacioso"
Don Quijote – Capítulo VI – Segunda Parte

Con la experiencia obtenida en el desarrollo del entorno inteligente (ver capítulo 3), basándose en el primer modelo de interacción mediante diálogos orales y los estudios y evaluación inicial (ver capítulo 1) y siguiendo las consideraciones establecidas sobre el reconocimiento de voz (ver apartado 3.9.2) se ha diseñado, desarrollado y evaluado (ver capítulo 1) un sistema de diálogos automático para interaccionar con las entidades del entorno inteligente.

Dadas las características heterogéneas y dinámicas de los entornos inteligentes, que se han especificado en diversas ocasiones anteriormente (ver capítulo 2), resulta fundamental que cada diálogo se pueda crear automáticamente, adaptándose a las características específicas de cada entorno. Para ello, junto a la definición del entorno y sus entidades (ver apartado 3.2), se define la información lingüística de cada una de ellas, la descripción del entorno y las gramáticas asociadas. Estos elementos se emplean para crear la interfaz de forma automática y adaptada a cada entorno.

El sistema de diálogos creado se basa en una estructura en árbol, donde las partes lingüísticas están representadas en nodos. Además presenta un conjunto de gramáticas creadas automáticamente que definen todas las posibles formas de interacción dentro del entorno (ver apartado 3.9.2.2).

Este árbol se utiliza en los procesos de comprensión e interacción con los usuarios, esto es, para realizar interpretación y generación de oraciones. La interfaz permite recuperarse de errores producidos por el reconocedor, procesar oraciones en donde se ha omitido parte de la información o resolver la anáfora pronominal. Una de las piezas claves en la realización de estas tareas es el continuo empleo de la información sobre el contexto físico almacenada en la pizarra (ver apartado 3.3). Dadas las características dinámicas de esta interfaz los procesos de interpretación y generación variarán automáticamente de un entorno a otro y se adaptarán al estado del entorno en cada momento.

Dentro del entorno inteligente implementado se ha definido una interfaz que permite interaccionar con las entidades que componen al mismo (ver apartado 3.5). Esta interacción se adapta de forma automática a las características del entorno y permite comunicarse con las entidades del mismo de forma natural. La Figura 23 corresponde a la trascripción de un diálogo real realizado en el entorno desarrollado, en donde se muestran algunas de las ideas que se plasmarán en este capítulo. La interacción se realiza con dos lámparas que poseen una luz halógena regulable y un foco de lectura cada una, los fluorescentes principales (un total de cinco luces posibles) y un sintonizador de radio. El resultado del reconocedor se representa entre llaves y los comentarios sobre la interacción en cursiva.

1. Usuario: Odisea {Odisea}

2. Sistema: ¿Si?

3. U: ¿Puedes encender la luz? {encender luz}

El número de luces posibles es de cinco, se reduce la respuesta generada

4. S: ¿Qué luz quieres encender?

El usuario tarda más de siete segundos en responder. El sistema le ofrece asistencia mostrándole todas las opciones posibles

5. S: Puedes encender la luz del techo, de lectura de la izquierda o de la derecha o de ambiente de la izquierda o de la derecha

6. U: La luz de la izquierda, por favor {luz izquierda}

Se salta un nodo en el árbol y realiza clarificación entre las dos luces posibles

7. S: ¿Prefieres encender la luz de lectura de la izquierda o la luz de ambiente de la izquierda?

8. U: La de ambiente {ambiente}

Se enciende la luz de ambiente de la izquierda

9. U: Quiero que enciendas la luz del techo {encender techo}

El reconocedor sólo ha devuelto parte de la información. Al sólo existir una entidad relativa al techo que actualmente se pueda encender el sistema se recupera del error y enciende la luz del techo

10. U: ¿Puedes encender la luz? {encender luz}

El usuario repite la oración del punto número 3. En esta ocasión, al haber sólo tres posibles entidades, para la misma entrada la oración de respuesta es distinta

11. S: ¿Prefieres de lectura de la izquierda o de ambiente de la izquierda o de la derecha?

12. U: La luz de la izquierda, por favor {luz izquierda}

El usuario repite la oración pronunciada en el punto 6. En este caso, al sólo existir una luz a la izquierda que esté apagada, el sistema no necesita que el usuario suministre toda la información y directamente enciende la luz de lectura de la izquierda

13. U: Apágala {apagarlo}

El usuario emplea una anáfora por lo que se ejecuta la acción sobre la última entidad. Se apaga la luz de lectura de la izquierda

14. U: ¿Podrías poner la radio? {poner radio}

El sistema llega a un subdiálogo y pronuncia la oración específica que lo continúa

15. S: ¿Qué emisora prefieres poner?

16. U: M80 radio, por favor {m_80)

Se enciende la radio con m80

17. U: Me gustaría que subieras el volumen de la radio {subir}

El reconocedor sólo es capaz de devolver subir. Al existir dos entidades que permiten realizar esta acción pregunta al usuario para saber sobre cuál realizarla

18. S: ¿Prefieres el volumen de la radio o la luz de ambiente de la izquierda?

19. U: La radio {radio}

El usuario responde sin proporcionar toda la información solicitada. El sistema interpreta la oración correctamente y sube el volumen de la radio

20. U: Más fuerte, por favor {más}

Se repite la última acción realizada: se vuelve a subir el volumen de la radio

21. U: Quítala {quitarlo}

Se resuelve la anáfora y se aplica la acción sobre la última entidad. Se apaga la radio

22. U: Quiero que apagues la luz de ambiente de la izquierda {luz ambiente izquierda}

El reconocedor no devuelve toda la oración. El sistema comprueba qué acciones se pueden realizar sobre esa entidad y responde ofreciéndolas

23. S: ¿Prefieres apagar, subir o bajar la luz de ambiente de la izquierda?

24. U: Prefiero que la apagues {apagar}

Se realiza la acción sobre la última entidad referida, esto es, el sistema apaga la luz de ambiente de la izquierda

25. U: Me gustaría que quitaras la luz {quitar luz)

Al existir sólo una luz encendida el sistema no necesita clarificación, como en casos anteriores. Directamente apaga la luz del techo

Figura 23. Trascripción de un diálogo real llevado a cabo en el entorno desarrollado

Las respuestas y acciones que realiza el sistema se generan de forma automática a partir de las entidades que se encuentran presentes en el entorno. Si la configuración del entorno fuera distinta la creación de la interfaz se adaptaría automáticamente al nuevo entorno, variando su comportamiento. Si para el ejemplo anterior, las lámparas no tuvieran las luces halógenas regulables (sólo hubiera tres luces posibles) las mismas solicitudes del usuario tendrían respuestas diferentes por parte del sistema. La Figura 24 repite las seis interacciones iniciales realizadas por el usuario para este nuevo entorno.

1. Usuario: Odisea {Odisea}

2. Sistema: ¿Si?

3. U: ¿Puedes encender la luz? {encender luz}

El número de luces posibles es de tres, se ofrecen todas las posibles luces que se pueden encender

4. S: ¿Prefieres encender la luz del techo, de lectura de la izquierda o de la derecha?

El usuario tarda más de siete segundos en responder. El sistema le ofrece asistencia volviendo a mostrarle las opciones posibles

5. S: Puedes encender la luz del techo, de lectura de la izquierda o de la derecha

6. U: La luz de la izquierda, por favor {luz izquierda}

Se salta un nodo en el árbol, al sólo existir una luz a la izquierda el sistema interpreta directamente que se refiere a la luz de lectura de la izquierda por lo que enciende esta luz

Figura 24. Trascripción de las primeras interacciones del mismo diálogo para el nuevo entorno

En este nuevo ejemplo, tras pronunciar las mimas oraciones el sistema proporciona respuestas diferentes en los puntos 4 y 5. Además, Tras la oración número 6, el sistema es capaz de realizar directamente una acción, en lugar de solicitar clarificación como ocurría en el caso anterior.

Las variaciones en el comportamiento de la interfaz de diálogos son automáticas, y se basan en la composición del entorno en el que se encuentra. Para ello es necesario

definir las entidades que forman parte del entorno y la información lingüística asociada a cada una de ellas.

5.1 Definición de la interfaz de interacción con el entorno

La interfaz de diálogos orales de interacción con el entorno se crea de forma automática a partir de la información contenida en el Documento de Descripción del Entorno (DDE, ver apartado 3.2). De la misma manera que se ha visto ocurría con otros elementos, la definición de la información lingüística se establece en los DPCE, relacionándola con las clases de entidad. De este modo las instancias de esos tipos heredan automáticamente todas las propiedades definidas.

La definición de la interfaz se compone de dos fases diferenciadas que se pueden realizar en momentos diferentes y por distintas personas:

- En primer lugar se deberá definir en los DPCE la información relacionada con cada nueva clase de entidad. Esta comprende la información lingüística de interacción, los métodos necesarios para la automatización del sistema y, opcionalmente, de nuevas plantillas de gramática. Esta información se definirá una única vez y será compartida por todas las entidades del mismo tipo.
- En segundo lugar se deben definir en los DDEE qué entidades se encuentran presentes en el entorno y de qué tipo son. La información lingüística de cada entidad del entorno vendrá dada por la que han establecido anteriormente las clases de entidad en los DPCE. De este modo es posible que para crear una interfaz de diálogos orales de interacción con el entorno sólo sea necesario definir los elementos que se encuentran en el mismo, sin necesidad de modificar o añadir ningún tipo de información lingüística adicional. Sin embargo, la información que hereda cada entidad también se puede personalizar, adaptándola a las características especiales del entorno.

5.1.1 Definición de la información lingüística asociada a las clases de entidad

Como se acaba de mencionar, cada clase de entidad tiene asociada su propia información lingüística que establece todas las posibles interacciones que se pueden realizar con las entidades de ese tipo. Esta información se clasifica en siete posibles partes lingüísticas, sin perjuicio de poder ser ampliadas fácilmente añadiendo nuevas partes. Las siete partes lingüísticas que se proponen son:

- Parte verbal (VP). Describe las acciones que se pueden llevar a cabo con la entidad.
- Parte objeto (OP). Establece los posibles nombres que pueden tomar las entidades que reciben la acción del verbo.
- Parte de ubicación (LP). Describe su situación física dentro del entorno.

- Parte de objeto indirecto (IOP). Especifica a quién o a qué va dirigida la acción que se realiza.
- Parte modal (MODALP). Indica el modo en que se debe realizar la acción.
- Parte cuantificadora (QP). Define un valor o cantidad que se aplica sobre la acción que se realiza.
- Parte modificadora (MP). Añade información calificativa a alguna de las partes anteriores.

La información lingüística asociada a las clases de entidades se define mediante los DPCE de la misma forma que se parametrizaban las propiedades y parámetros de la definición de clases (ver apartado 3.2.1.1). Esta información se compartirá posteriormente por todas las entidades de ese tipo que se definan en el entorno. Esto es, dos entidades del mismo tipo heredan la misma información lingüística que posteriormente puede ser personalizada, dependiendo de las características propias del entorno.

```
<classes>
   definición_clase
   [, definición_clase ] ...
</classes>

definición_clase:
<class name="nombre">
   <property name="propiedad">
      <paramSet name="dialogue">

         <param name="action1"> Nombre_de_la_acción </param>
         <param name="skeleton1"> Parte Palabra [, Parte Palabra ]...
         </param>
         [, <param name="skeleton2"> Parte Palabra
         [, Parte Palabra ]...
            </param> ] ...
         [, <param name="action2"> Nombre_de_la_acción </param>
            <param name="skeletonn"> Parte Palabra
         [, Parte Palabra ]...
            </param>
            [, <param name="skeletonm"> Parte Palabra
         [, Parte Palabra ]...
               </param> ] ... ]...

            </paramSet>
   </property>
</class>

Parte:
VP | OP | LP | IOP | MODALP | QP | MP
```

Figura 25. Información lingüística que se adjunta en el DPCE a la clase de entidad

Para añadir la información referente a la interfaz lingüística se ha de adjuntar un conjunto de parámetros bajo la etiqueta de nombre *dialogue*. A continuación habrá un parámetro *action* por cada una de las posibles acciones que se pueden realizar con la entidad y, para cada uno de ellos uno o varios parámetros *skeleton* que definen los posibles esqueletos de oraciones que se pueden invocar para realizar esa acción. La sintaxis se muestra en la Figura 25.

Los parámetros *action* identifican una nueva acción y contienen la descripción del tipo de acción que se realiza. Los esqueletos de oraciones se identifican mediante el parámetro *skeleton* y contienen las palabras claves que los constituyen y que permiten realizar la acción que tienen asociada. Cada palabra clave debe ir precedida por la parte lingüística que representa (*VP, OP, LP, IOP, MODALP, QP* o *MP*). Si se desea especificar sinónimos de una misma parte lingüística se pueden escribir dos o más palabras clave seguidas. Los esqueletos de oraciones pueden iniciarse con cualquier parte lingüística y éstas se pueden repetir cuantas veces sea necesario.

```
<classes>
   <class name="dimmerlight">

      <property name="status">
         <paramSet name="dialogue">
            <param name="action1">encender</param>
            <param name="skeleton1">VP encender poner OP luz lámpara
            MP ambiente halógena</param>
            <param name="action2">apagar</param>
            <param name="skeleton2">VP apagar quitar OP luz lámpara
            MP ambiente halógena</param>
         </paramSet>
      </property>

      <property name="value">
         <paramSet name="dialogue">
            <param name="action1">subir</param>
            <param name="skeleton1">VP subir aumentar OP luz lámpara
            MP ambiente halógena</param>
            <param name="skeleton2">VP subir aumentar OP intensidad
            LP luz lámpara MP ambiente halógena</param>
            <param name="action2">bajar</param>
            <param name="skeleton3">VP bajar disminuir reducir
            OP luz lámpara MP ambiente halógena</param>
            <param name="skeleton4">VP bajar disminuir OP intensidad
            LP luz lámpara MP ambiente halógena</param>
         </paramSet>
      </property>

   </class>
</classes>
```

Figura 26. Ejemplo de la definición lingüística de una entidad de tipo luz regulable

En la Figura 26 se muestra la información lingüística asociada a una entidad de tipo *luz regulable* o *dimmerLight* (ya utilizada en el apartado 3.2). En este caso existen cuatro posibles acciones que se pueden realizar sobre una luz regulable: *encender, apagar, subir y bajar*. Para las acciones de *encender* y *apagar* se especifica un esqueleto de oración con tres partes lingüísticas. Cada parte contiene dos sinónimos, por lo que existen ocho posibilidades distintas para poder encender una luz regulable y otras ocho para apagarla. En el caso de las acciones de *subir* y *bajar* se emplean dos esqueletos de oraciones, ya que cada una se compone de partes distintas. Existen dieciséis posibilidades para realizar cada una de estas acciones. Además, hay que considerar que no es necesario utilizar todas las partes en todo momento (dependiendo de las circunstancias propias de cada entorno y su contexto se necesitará una, varias o todas, ver más adelante el apartado 5.3.3) y que el empleo de gramáticas permite un uso flexible de los esqueletos de oraciones (ver más adelante el apartado 5.2.1), por lo que el número de posibilidades se incrementa considerablemente. Los esqueletos de las acciones *encender* y *apagar* están asociados a la propiedad *status* mientras que los esqueletos de las acciones *subir* y *bajar* quedan asociados a la propiedad *value*.

Este documento se puede editar y modificar para adaptarlo a distintas formas orales de interacción. Por ejemplo, se podrían añadir, quitar o cambiar algunos de los sinónimos o de las partes para adaptar la interacción a modos regionales o transnacionales de interaccionar, nombrar o dirigirse a las entidades. Se ha buscado que los modos de interacción sean fácilmente definibles y reconfigurables, pudiendo cambiar o crecer fácilmente. No es necesario modificar la implementación de la interfaz de diálogos, sino que basta con editar y modificar las definiciones establecidas en el DPCE.

5.1.1.1 Definición de subdiálogos

Además de los subdiálogos que el sistema produce de forma automática durante el proceso de interacción con el usuario (ver más adelante el apartado 5.3.3), los esqueletos de oraciones permiten definir subdiálogos específicos que se producen dentro de la realización de la misma acción. Estos subdiálogos se suelen utilizar cuando se quiere informar o preguntar al usuario con oraciones concretas (en lugar de utilizar la respuesta generada automáticamente por el sistema, ver más adelante el apartado 5.3.4). La definición de subdiálogo se establece utilizando las letras *SUB* (por *subdialogue*), seguidas de la parte o partes que lo forman. Esta definición se puede establecer de forma opcional delante de la definición de cualquier parte lingüística (exceptuando la primera) y en tantas ocasiones como subdiálogos se deseen crear. La sintaxis de definición de subdiálogos se muestra en la Figura 27.

```
<paramSet name="dialogue">

   <param name="action1"> Nombre_de_la_acción </param>
   <param name="skeleton1">Parte Palabra [, [SUB] Parte Palabra ]...
   </param>
   [, <param name="skeleton2">Parte Palabra [, [SUB] Parte Palabra
]...
      </param> ] ...
   [, <param name="action2"> Nombre_de_la_acción </param>
      <param name="skeletonn">Parte Palabra [, [SUB] Parte Palabra
]...
      </param>
      [, <param name="skeletonm">Parte Palabra
         [, [SUB] Parte Palabra ]...
         </param> ] ... ]...

</paramSet>
```

Figura 27. Definición de la información lingüística de una acción con subdiálogos

Un caso de estas características se produce con el diálogo asociado a las entidades de tipo *radio*. Si tras una oración de la forma *Por favor, podrías poner la radio* se quiere que el sistema responda una pregunta concreta, como por ejemplo *¿Qué emisora prefieres poner?*, sería necesario establecer una definición de subdiálogos. En la Figura 28 se muestra este subdiálogo para tres emisoras: M80 (definida en la gramática como m_80), Radio 5 (definida como radio_5) y los 40 Principales (definida como cuarenta_principales y cuarenta).

```
<paramSet name="dialogue">

   <param name="action1">encender_m_80</param>
   <param name="skeleton1"> VP encender poner OP radio SUB MP m_80
   </param>

   <param name="action2">encender_radio_5</param>
   <param name="skeleton2"> VP encender poner OP radio SUB MP radio_5
   </param>

   <param name="action3">encender_cuarenta</param>
   <param name="skeleton3"> VP encender poner OP radio
   SUB MP cuarenta_principales cuarenta </param>

</paramSet>
```

Figura 28. Diálogo para encender la radio con tres posibles subdiálogos

Este subdiálogo permite que el usuario pueda pronunciar oraciones del tipo *Quiero encender la radio* y que, tras una respuesta generada específicamente en el sistema (ver más adelante el apartado 5.3.3.1), pueda pronunciar *Los cuarenta principales* y así se

complete la acción. También estaría permitido completar los subdiálogos con una única oración, por lo que la frase *Quiero encender la radio con los cuarenta principales* tendría el mismo resultado. Finalmente, oraciones del tipo *Pon los cuarenta principales*, donde no se presenta toda la información, también tienen el mismo efecto que las anteriores, ver más adelante el apartado 5.3.3.6.

5.1.2 Definición de las instancias de entidad

Una vez definida la información lingüística asociada las clases de entidades, y dado que todas las entidades del mismo tipo comparten las mismas posibilidades de interacción, en muchas ocasiones basta con definir qué entidades se encuentran en el entorno para que automáticamente se cree una interfaz oral adaptada al mismo.

La definición de los elementos que se encuentran en el entorno se especifica en los DDEE (ver apartado 3.2.2 y apartado 3.2.3). La Figura 29 vuelve a mostrar la sintaxis de la definición de las entidades.

```
<instances>
   definición_instancia
   [, definición_instancia ] ...
</instances>

definición_instancia:
<entity name="nombre" type="clase_entidad"/>
[, <entity name="nombre" type="clase_entidad"/> ] ...
```

Figura 29. Sintaxis del DDEE

Sin embargo, en ocasiones será necesario (o simplemente recomendable) especificar información lingüística relativa al entorno específico sobre el que se crea la interfaz. Este es el caso, por ejemplo, de un entorno donde aparezcan varias entidades del mismo tipo, en donde será necesario añadir nueva información que permita diferenciar entre ellas. También se produce esta circunstancia cuando existen características particulares de la entidad en el entorno, que no se pueden especificar en una definición general del tipo de entidad (tal es el caso del color, tamaño, posición, etc.).

Para solventar estas circunstancias, además de la simple definición de las entidades que están presentes en el entorno, se utilizan los DPEE que permiten especificar nueva información lingüística concreta para cada entidad. Esto se realiza empleando dentro del conjunto *dialogue* el parámetro *add* en la propiedad donde se quiere añadir la nueva información. A continuación, el atributo debe ir seguido por un número, que señala un esqueleto de oración concreto al que añadir la nueva información o por la palabra *all*, que especifica que la información se debe añadir a todos esqueletos de oración de esa propiedad. La Figura 30 muestra la sintaxis de la parametrización lingüística de la propiedad de una entidad.

Un ejemplo de cómo se puede añadir información lingüística adicional se representa en la Figura 31. Para la propiedad *status* la información lingüística adicional se adjunta en todos sus esqueletos de oraciones. Para la propiedad *value* sólo se añade la información lingüística (que además es diferente) en los esqueletos de oraciones primero y tercero.

```
<entity name="nombre">
   <property name="propiedad">
      <paramSet name="dialogue">

         <param name="add_(1, all)">valor</param>
         [ , <param name="add_2">valor</param> ] ...

      </paramSet>
   </property>
</entity>
```

Figura 30. Sintaxis de la parametrización lingüística en un DPEE

```
<instances>
   <entity name="lampv1">

      <property name="status">
         <paramSet name="dialogue">
            <param name="add_all">LP izquierda</param>
         </paramSet>
      </property>

      <property name="value">
         <paramSet name="dialogue">
            <param name="add_1">LP izquierda</param>
            <param name="add_3">LP izquierdo</param>
         </paramSet>
      </property>

   </entity>
</instances>
```

Figura 31. DPEE al que se añade información lingüística

Nuevamente, la idea fundamental es permitir configurar, describir, modificar y adaptar de forma sencilla las interacciones lingüísticas que se pueden llevar a cabo con las entidades del entorno. Una vez definida en los DPCE la información lingüística asociada a las clases de entidades (posiblemente por otra u otras personas) un diseñador de la interfaz del entorno sólo necesita definir mediante los DDEE qué entidades están presentes en el entorno. Este u otros diseñadores especializados podrán a su vez, por medio de los DPEE, modificar estas definiciones, adaptando la interfaz a las características propias de interacción del entorno determinado. Toda esta información

se une en el Documento de Descripción de la Entidad (DDE), que sirve de soporte para la creación automática de la interfaz de diálogos orales.

La definición de la información lingüística se realiza de forma análoga a la que permite definir las entidades, sus parámetros y otras interfaces (ver apartado 3.2). De este modo se consigue una forma estándar y homogénea de definición del entorno, sus elementos y posibilidades multimodales (por ejemplo, mediante interfaz Web y de diálogos orales) de interacción.

5.1.3 Definición de los métodos asociados a los tipos de entidad

Cuando se define una entidad, además de aportar la información lingüística asociada, se tienen que implementar dos métodos que permitan la interacción automática de las entidades con el entorno. Estos métodos son el método *BBAction* y *hasRequestedADifferentAction*. Ambos serán comunes para todas las entidades del mismo tipo, por lo que sólo será necesario implementarlos cuando se defina un nuevo tipo de entidad. La implementación específica y adecuada de estos métodos se asocia automáticamente a cada entidad de ese tipo tras la definición de las entidades en el entorno (ver apartado 5.1.2).

5.1.3.1 El método BBAction

Este método define qué acción se debe realizar con la entidad dependiendo de la solicitud del usuario. Como entrada recibe el elemento de la pizarra sobre el que se debe interaccionar (definido en el DPCE tras el atributo *name*, ver Figura 30 y Figura 31 del apartado 5.1.2), el tipo de acción que corresponde con la solicitud del usuario (definido en el DPEE tras el atributo *action*, ver Figura 27 y Figura 28 del apartado 5.1.1) y la oración pronunciada por el usuario. Con esta información el método puede realizar una acción dentro del entorno (utilizando para ello la pizarra, ver apartado 3.3) o puede preparar una respuesta para el usuario (ver apartado 5.3.3.1).

Por ejemplo, basándonos en la información lingüística de una entidad de tipo *DimmerLight* (ver Figura 26 y Figura 30), si un usuario pronuncia la oración *por favor, quiero que subas la luz de ambiente* el método *BBAction* recibiría que esta entidad interacciona con el elemento *lamp_1* de la pizarra, que la acción que se ha solicitado es *subir* y que la oración pronunciada por el usuario es *subir luz ambiente* (ver más adelante el apartado 5.2.1). Con esos datos el método podrá ejecutar la acción física solicitada utilizando la entidad correspondiente de la pizarra (la información sobre la oración pronunciada por el usuario no es estrictamente necesaria, pero se recibe como posible soporte en la toma de decisiones sobre qué acción realizar).

Este método también se invoca cuando se completa un subdiálogo. Por lo tanto, tal y como se especificaba en el apartado 5.1.1.1, si se quiere reproducir mediante síntesis de voz un mensaje específico en ese punto, este método es el encargado de realizar esta tarea.

Aquellos tipos de entidad sencillos, que realicen acciones binarias del tipo 1/0 (como *encender* y *apagar*), o los que heredan de otros tipos de entidad con las mismas características, no necesitan definir este método, ya que el sistema proporciona uno por defecto en el primer caso u obtienen el de la entidad de la que se hereda en el segundo.

5.1.3.2 El método hasRequestedADifferentAction

Este método recibe los mismos argumentos que el anterior y devuelve verdadero o falso dependiendo de si el estado actual de la entidad es distinto o no al que se requiere en la frase pronunciada. Para ello el método consulta el estado de la entidad en la pizarra y lo compara con el requerido. En el caso de que ambos estados sean distintos el método devuelve verdadero, en caso contrario devuelve falso.

Por ejemplo, supongamos que la entidad *lamp_1* que es de tipo *DimmerLight* se encuentra apagada dentro del entorno. Si el método recibe que la acción solicitada es *apagar* y que la oración pronunciada fue *apagar luz ambiente*, tras comprobar el estado físico de esta luz en la pizarra devolverá falso, ya que su situación actual y la información recibida es la misma.

Este método se utiliza como elemento imprescindible para automatizar el proceso de interpretación de las oraciones pronunciadas por el usuario. La información sobre su uso se explica más adelante en los apartados 5.3.3.2 y 5.3.3.3.

Nuevamente, los tipos de entidad sencillos que realicen acciones binarias o los que hereden de otros tipos de entidad con las mismas características no necesitan definir este método, ya que el sistema proporciona uno por defecto, en el primer caso, o se emplea el de la entidad de la que se hereda en el segundo.

Como se puede ver, ambos métodos son generales y se pueden emplear por igual con cualquier otra entidad del mismo tipo. Tienen que ser implementados conjuntamente con el diseño de la información de interacción con los tipos de entidad y no es necesaria su modificación por parte de la persona que diseña la interfaz para un entorno determinado.

5.1.4 Definición de plantillas de gramáticas

El sistema de diálogos incorpora una plantilla de gramática que el sistema utiliza en el proceso de creación automático de la gramática asociada a cada entidad (ver más adelante el apartado 5.2.1). Esta gramática se basa en el JSGF, tal y como se especificó en el apartado 3.9.2.2).

La plantilla definida es lo suficientemente básica y general como para cubrir todas las posibles interacciones que se pueden realizar dentro de un entorno inteligente. Estas pueden variar desde realizar simples saludos o despedidas hasta interaccionar con las luces o controlar el aire acondicionado. La plantilla de gramática proporcionada incorpora una única regla pública, *utterance*, que contiene las reglas necesarias para

definir las múltiples oraciones que se pueden producir en un entorno. Las oraciones definidas dentro de esta plantilla de gramática son:

- Oraciones nominales. Como por ejemplo *Prefiero la de la izquierda* o *La lámpara.*
- Oraciones en presente del subjuntivo. Como Quiero que se encienda la luz o Que se apaguen los fluorescentes.
- Oraciones en pasado del subjuntivo. Por ejemplo *Me gustaría que se apagara el aire acondicionado.*
- Oraciones imperativas. Es el caso de *Sube el volumen de la radio* o *Ábrela.*
- Oraciones interrogativas. Como por ejemplo *¿Puedes llamar a Javier?.*

La plantilla de gramática definida se compone de numerosas reglas no terminales y de un conjunto de reglas terminales vacías que han de estar presentes en la gramática de forma obligatoria. El conjunto de estas reglas terminales obligatorias se especifica en la Figura 32.

```
Male imperative informal verb with pronoun
Male imperative formal verb with pronoun
Female imperative informal verb with pronoun
Female imperative formal verb with pronoun
Imperative informal verb
Imperative formal verb
Infinitive verb
Subjunctive present plural verb
Subjunctive present singular informal verb
Subjunctive present singular formal verb
Subjunctive past singular verb
Subjunctive past plural verb
Singular female noun, singular male noun
Plural female noun
Plural male noun
Invariant common noun
Singular male modifier noun
Singular female modifier noun
Plural male modifier noun
Plural female modifier noun
Invariant common modifier noun
Modifier adverb
```

Figura 32. Reglas terminales que deben estar presentes en toda plantilla de gramática

Posteriormente, durante el proceso de creación automático de la gramática de la entidad (ver apartado 5.2.1), algunas de estas reglas terminales obligatorias se rellenarán y otras permanecerán vacías.

Como se puede comprobar la plantilla define numerosas palabras, modos y combinaciones opcionales u obligatorias que hacen que se aumente considerablemente

la naturalidad en la interacción con el entorno. Por ejemplo, la Figura 33 muestra la composición de la regla de oraciones en pasado del subjuntivo con otras reglas no terminales.

```
<subjunctive past sentence> = ["me gustaría" | quisiera | desearía |
preferiría] que se (<subjunctive past singular verb> <singular noun> |
<subjunctive past plural verb> <plural noun>);
```

Figura 33. Definición de una de las reglas de la plantilla

Además de la plantilla de gramática ya definida en el sistema, es posible definir otras plantillas de gramáticas distintas y asociarlas a tipos de entidades determinados. De este modo, durante el proceso de creación automático de la interfaz las entidades de ese tipo utilizarían la nueva plantilla definida en lugar de la plantilla proporcionada por el sistema.

Cualquier nueva plantilla de gramática que se defina puede tener tantas reglas no terminales como se desee y éstas pueden recibir cualquier nombre. La única condición necesaria es que la nueva plantilla contenga el conjunto completo de reglas terminales vacías obligatorias mostrado en la Figura 32 y que éstas conserven el mismo nombre.

Nuevamente, un diseñador de la interfaz para un entorno no necesita preocuparse de qué plantilla de gramática ha de utilizar pudiendo en la inmensa mayoría de los casos utilizar la plantilla de gramática definida por defecto. Sin embargo, en caso de que considerarlo necesario, podría definir nuevas plantillas de gramáticas que se adapten mejor a las necesidades concretas de un entorno.

5.2 Creación automática de la interfaz de diálogos orales

Con la información lingüística obtenida del DDE (ver apartados 3.2 y 5.1) se crea de forma automática la interfaz de diálogos orales adaptada a las características concretas del entorno.

El proceso de creación se basa en dos pasos que se realizan de forma paralela. El primero consiste en la creación de un conjunto de gramáticas adecuadas para la interacción con el entorno (ver apartado 3.9.2.2). El segundo se basa en la construcción de un árbol lingüístico que será utilizado en los procesos de interpretación y generación (ver más adelante el apartado 5.3).

En el proceso de creación de la interfaz el sistema lee el DDE. Para cada una de las entidades representadas en el documento (esto es, aquellas que están presentes en el entorno) se obtiene la información lingüística asociada a la clase de entidad (ver apartado 5.1.1) y, si es necesario, se añade la nueva información lingüística representada en el DDE que es específica para esa entidad. Empleando esta información y los elementos de la pizarra con los que interactúa cada entidad se construyen las gramáticas y el árbol de representación lingüística.

5.2.1 Creación del conjunto de gramáticas del sistema

El sistema presentará una gramática por cada tipo de entidad diferente. Esto quiere decir que varias entidades del mismo tipo comparten la misma gramática. Las gramáticas se basan en la plantilla de gramática definida en el apartado 5.1.4, a no ser que se haya definido una nueva plantilla específica para ese tipo de entidades.

Tras obtener toda la información lingüística asociada a la entidad se comprueba si ya ha sido creada una gramática para ese tipo de entidad. En caso contrario se empieza el proceso a partir de una plantilla vacía. Si una entidad del mismo tipo ya ha creado la gramática anteriormente, se continúa añadiendo información a esta misma gramática.

Con cada una de las palabras que componen la información lingüística de la entidad se realiza un análisis sintáctico. Para el análisis sintáctico se utilizan las herramientas lingüísticas Maco+ y Relax [Carmona, J. et al 1998], desarrolladas por el grupo de investigación en Procesamiento del Lenguaje Natural de la Universidad Politécnica de Catalunya y el Laboratorio de Lingüística Computacional de la Universidad de Barcelona. Estas herramientas devuelven una anotación morfosintáctica de las palabras utilizando etiquetas PAROLE (un conjunto homologado de etiquetas morfosintácticas para varias lenguas europeas) [Martí, M.A. et al. 1998].

Un ejemplo de los resultados obtenidos se muestra en la Figura 34. La palabra *encender* es un verbo principal en infinitivo, *luz* es un nombre común en femenino singular y *techo* un nombre común en masculino singular.

```
encender VMN0000
luz NCFS000
techo NCMS000
```

Figura 34. Etiquetas morfosintácticas para las palabras encender, luz y techo

En algunos casos el analizador podría obtener un análisis incorrecto de la palabra procesada o podría ser necesario desambiguar palabras con múltiples posibilidades de análisis. Para simplificar esta tarea, en aquellas palabras que se desee, se puede realizar un etiquetado manual, evitando tener que enviar la palabra al analizador. Para realizar esto basta con anteceder la etiqueta morfosintáctica a la palabra que se desee en la definición del esqueleto de la oración. Basándonos en el ejemplo de la Figura 26 se podría especificar la forma

```
VP encender poner OP NCFS000:luz lámpara MP ambiente halógena
```

de modo que se analizarían todas las palabras menos *luz*, que ya posee su etiquetado morfosintáctico. Además, para evitar análisis reiterativos, antes de enviar la palabra a los analizadores se comprueba si ya se han procesado anteriormente. En caso de que así sea se utiliza la misma etiqueta morfosintáctica del primer análisis realizado.

Una vez analizada la palabra se rellenan las partes necesarias de la gramática asociada. Para esto se tiene en cuenta además si se trata de una parte verbal, objeto, modificadora, etc. (ver apartado 5.1.1). Por ejemplo, en el caso de un verbo se obtienen los modos y formas verbales del verbo especificado y se añaden a reglas tales como *imperative formal verb, subjunctive present singular formal verb, subjunctive present singular informal verb*, etc. obteniendo así todas las posibles formas que se pueden emplear con el verbo dado. Si por el contrario el análisis hubiera devuelto como resultado un nombre femenino se añadiría a la regla *singular female noun* en el caso de que fuera una parte objeto y a la regla *singular female modifier noun* en el caso de ser una parte modificadora.

Además, junto a la palabra añadida se escribe la etiqueta correspondiente a la palabra original que aparecía en el esqueleto de oración de la información lingüística asociada al tipo de entidad. El contenido de esta etiqueta es el que recibirá la aplicación como resultado del reconocimiento de voz, simplificando el procesamiento de la información recibida por la interfaz (ver apartado 3.9.2.2). La Figura 35 muestra cómo quedarían las reglas de *nombre femenino singular* y *nombre modificador masculino singular* basándose en los esqueletos de oraciones representados en la Figura 26.

```
<singular female noun> = luz {luz} | lámpara {lámpara}
                         | intensidad  {intensidad};
<singular male modifier noun> = ambiente {ambiente};
```

Figura 35. Ejemplo de reglas de gramática construidas automáticamente

Por último, por cada uno de los verbos que se añaden a la gramática se obtiene la forma de referencia pronominal del verbo y se añade a las reglas de gramáticas correspondientes. Por ejemplo para el verbo *encender* también se añaden a las gramáticas las formas *encenderla, encenderlo, enciéndela, enciéndelo, enciéndala* y *enciéndalo*. Esto permite reconocer las formas verbales que podrán ser utilizadas para la resolución de la anáfora pronominal (ver más adelante el apartado 5.3.3.8).

La interfaz se compondrá de tantas gramáticas como tipos diferentes de entidades se hayan especificado en el entorno. Adicionalmente, se cuenta con una gramática de sistema que exclusivamente contiene la palabra que hay que invocar para despertar al sistema (en este caso la palabra *Odisea*, ver apartado 3.9.2.3). Posteriormente, durante el proceso de interacción con los usuarios, la gramática de sistema es la única que se encuentra activa mientras el sistema está dormido. Cuando un usuario lo despierta invocándolo por su nombre, la gramática de sistema se desactiva y se activan automáticamente las gramáticas descritas en el presente apartado, permitiendo establecer interacciones con el entorno. Cuando el sistema vuelve a dormir se desactivan todas las gramáticas activándose exclusivamente la gramática de sistema.

5.2.2 Creación del árbol lingüístico de interpretación y generación

Al mismo tiempo que se crean las gramáticas de interacción con las entidades del entorno se construye automáticamente un árbol lingüístico, que será un elemento fundamental en los procesos de interpretación y generación (ver más adelante el apartado 5.3).

El procedimiento de construcción parte de un nodo raíz vacío. Cada una de las partes de los esqueletos de oraciones asociados a las entidades (ver apartado 5.1) se convierte en información que se añade al árbol, bien sea en forma de un nuevo nodo o completando la información de un nodo existente. Cada nuevo nodo se compone de:

- La palabra y el tipo de parte especificados en el esqueleto de la oración.
- La lista de entidades de la pizarra que contiene esa palabra a ese mismo nivel.
- Información sobre si, para alguna entidad, esa palabra inicia un subdiálogo o no.
- Información que determina si el nodo se encuentra en un estado habilitado o deshabilitado.
- La lista de los nombres de las acciones asociadas a cada entidad (este nombre viene dado por el valor de los parámetros *action* en la definición de la información lingüística de las clases de entidad, ver el apartado 5.1.1).

Cuando el sistema se encuentra una parte donde aparecen varios sinónimos crea un nodo con cada uno de éstos. A partir de ese momento, los restantes hijos de ese mismo esqueleto de oración colgarán de cada uno de los sinónimos.

Supongamos, por ejemplo, que se declarara en el entorno una entidad llamada *lamp1* y que es de tipo *luz regulable* (ver declaración de entidades del entorno en los apartados 3.2 y 5.1). Basándose en la información lingüística representada en la Figura 26, tras procesar la primera línea correspondiente a su esqueleto de oración se generaría el árbol mostrado en la Figura 36.

Cada nodo difiere en el tipo de parte y palabra (como VP: encender o MP: ambiente) pero todos comparten que no inician ningún subdiálogo, se encuentran habilitados, la única entidad de la pizarra que está relacionada con cada nodo es *lamp1* y las acciones que los nodos representan sobre esas entidades son siempre *encender*.

A continuación, con el resto de esqueletos de oraciones de ese tipo de entidad se continuaría la construcción del árbol hasta completar todos los nodos que representan el conjunto de posibles interacciones y acciones que se pueden llevar a cabo con la entidad *lamp1*.

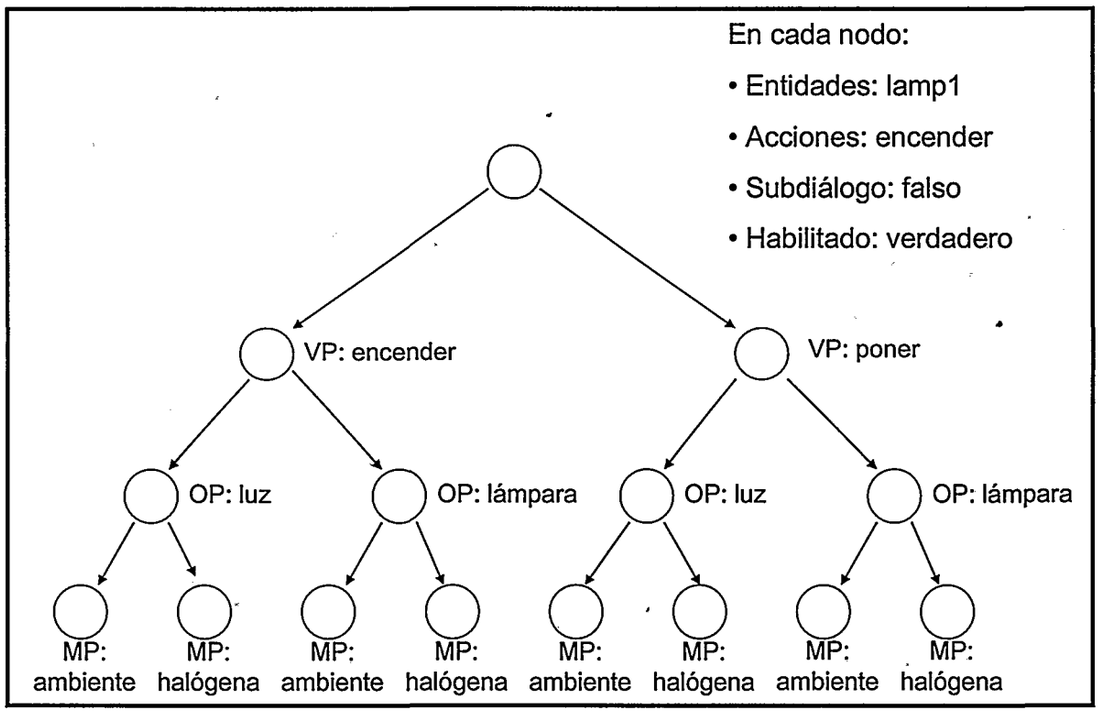


Figura 36. Árbol generado con un esqueleto de oración

Si dentro del entorno se declara una nueva entidad ésta empieza a añadir su información en el árbol desde la raíz. Si encuentra nodos con la misma palabra al mismo nivel, en lugar de crear un nuevo nodo añade su entidad correspondiente a la lista de entidades, la acción que representa a la lista de acciones y la información de si genera o no un subdiálogo. En caso contrario, crea un nuevo nodo tal y como se acaba de especificar en el paso anterior.

Supongamos que en el entorno se declara una nueva entidad denominada *fluorescent1*. Esta entidad es del tipo *luz fluorescente* y, entre otros, presenta los esqueletos de oraciones para la acción *encender* representados en la Figura 37 (simplificados para la mejor comprensión del ejemplo).

En este caso la parte verbal coincide con la parte verbal añadida al árbol por la entidad *lamp1*. Además, la parte objeto también coincide con uno de los nodos ya añadidos al árbol anteriormente. En estos casos no se crearán nuevos nodos sino que se añadirá información adicional a los nodos que ya se encuentran presentes en el árbol. Para el resto de las partes sí se crean nuevos nodos con la información correspondiente a la entidad *fluorescent1*.

Eliminando (con el fin de mejorar el visionado del resultado) los nodos de la parte derecha de la Figura 36, el árbol resultante sería el representado en la Figura 38.

```
<classes>
   <class name="fluorescent1">

      <property name="status">
         <paramSet name="dialogue">
            <param name="action1">encender</param>
            <param name="skeleton1">VP encender OP luz LP techo /param>
            <param name="skeleton2">VP encender OP fluorescente /param>
         </paramSet>
      </property>

   </class>
</classes>
```

Figura 37. DPCE con información lingüística para la acción encender de una entidad de tipo luz fluorescente

En este caso la información que comparten todos los nodos es que ninguno genera un subdiálogo, todos se encuentran habilitados y todos representan la acción de *encender*. Las diferencias se producen en la lista de entidades de pizarra relacionadas con cada nodo:

- En el caso de los nodos representados con un 1, las entidades de la pizarra con las que tienen relación son *lamp1* y *fluorescent1*.
- Los nodos representados con un 2 presentan relación con la entidad de la pizarra *lamp1*.
- Los nodos que aparecen con un 3 están relacionados con la entidad de la pizarra *fluorescent1*.

Se puede comprobar que el nodo *luz* sólo cuelga de *techo* y no de *lámpara*, a pesar de que *luz* y *lámpara* eran consideradas sinónimos por las entidades de tipo *luz regulable*. Sin embargo, no ocurre así con las entidades de tipo *luz fluorescente*. Esto quiere decir que, en este caso particular, se esperan y permiten interacciones del tipo *encender luz ambiente*, *encender lámpara ambiente* o *encender luz techo*, pero nunca interacciones del tipo *encender lámpara fluorescente*.

Este proceso se repite con cada uno de los esqueletos de oraciones de cada entidad que esté definida en el entorno, hasta crear un árbol completo que representa todas las posibles interacciones que se pueden llevar a cabo.

Cada ruta desde la raíz del árbol hasta una hoja representa uno de los diálogos que puede iniciar un usuario para realizar una tarea en el entorno. Los nodos intermedios representan información necesaria para poder llegar a realizar una acción (aunque como se verá más adelante esta información se puede obtener automáticamente a través del contexto, durante el proceso de interpretación). Los nodos hoja representan *nodos de acción*. Una vez alcanzados estos nodos se ha completado la información necesaria para

realizar una tarea y se puede realizar una acción en el entorno. Los nodos intermedios que especifican subdiálogos también representan *nodos de acción*.

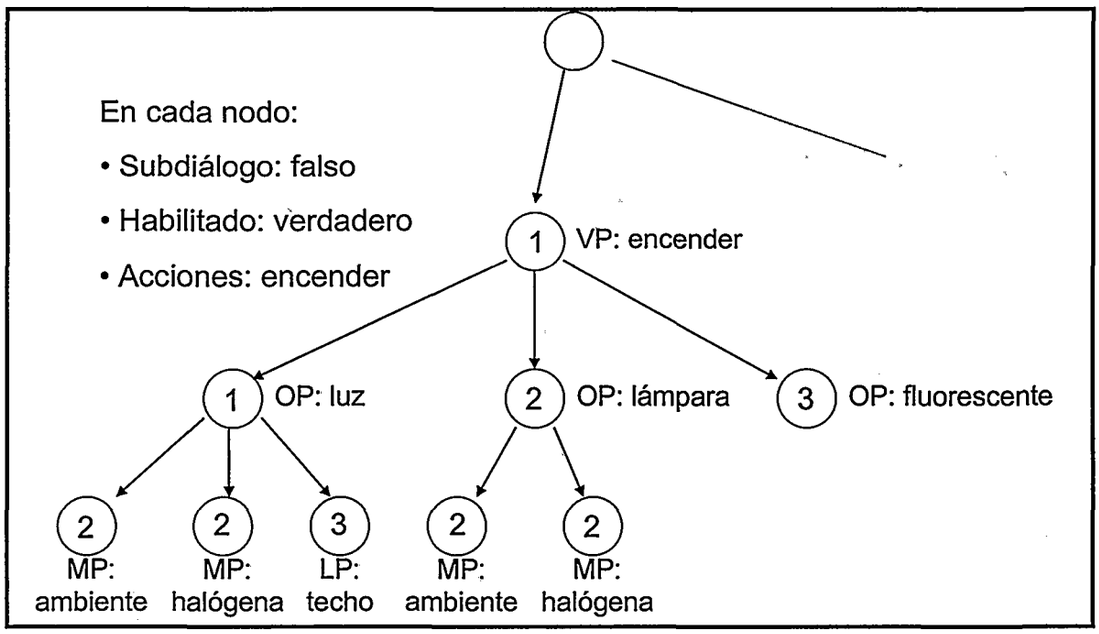


Figura 38. Parte del nuevo árbol generado al añadir la información de otra entidad

Como se puede comprobar en la Figura 38 no todos los nodos al mismo nivel tienen que corresponder con la misma parte lingüística. A su vez, los nodos del primer nivel no tienen necesariamente que pertenecer a una *parte verbal* (VP). Además, no todos los *nodos de acción* tienen que corresponder con una entidad física del entorno. En la Figura 39 se muestra el mismo árbol al que se le han añadido dos nuevos nodos que corresponden con una entidad de saludo (como un ejemplo simplificado del entorno real desarrollado). Estos nodos, que cuelgan de la raíz del árbol, corresponden con partes objeto y producen una respuesta del sintetizador cuando se ejecuta la acción que llevan asociada.

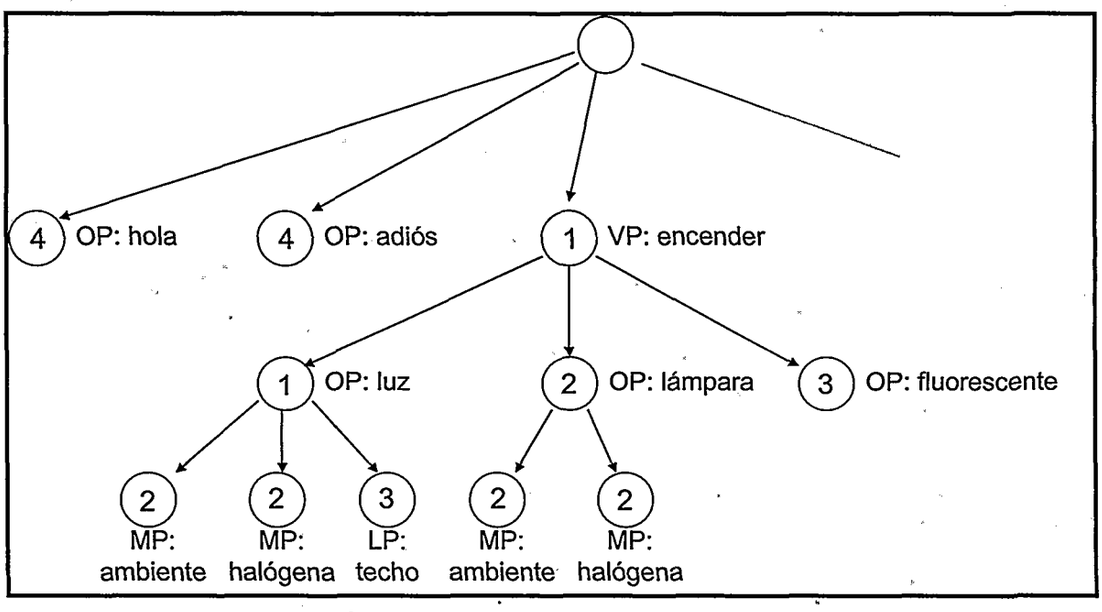


Figura 39. Árbol con nuevos nodos de una entidad de saludo

Los nuevos nodos, marcados con un 4, están a la misma altura del nodo *encender* y son *nodos de acción*. En este caso no tendrán como resultado la realización de una acción física en el entorno, sino que producirán una respuesta mediante el sintetizador de voz.

La creación automática de este árbol sólo se realiza una única vez basándose en las entidades del entorno (ver apartado 5.1.2) y la información lingüística proporcionada para cada clase de entidad (ver apartado 5.1.1). A partir de ese momento se puede iniciar la interacción con el entorno.

5.3 Interpretación y generación automáticas

Una vez construidos el árbol lingüístico y el conjunto de gramáticas los usuarios pueden empezar a establecer diálogos orales con las entidades del entorno.

Cuando el usuario inicia la interacción con el entorno el sistema recibe la salida del reconocedor de voz e intenta interpretar lo que se ha requerido. En el caso de que sea posible llevará a cabo una acción dentro del entorno. En caso contrario, deberá generar una respuesta oral que permita clarificar qué acción desea realizar el usuario.

Dado que la generación de la interfaz consiste en un proceso automático que se realiza dependiendo de las características concretas del entorno, la interpretación y generación también han de adaptarse de forma automática a cada entorno dado. Los procesos de interpretación y generación, para una misma solicitud del usuario, variarán

dependiendo de las entidades presentes en el entorno, el historial de interacción y el contexto físico del espacio.

El sistema de diálogos está gestionado por un supervisor que, tras recibir una oración del reconocedor (ver apartado 3.9.2.3), recorre el árbol interpretando la oración recibida y generando una posible respuesta. Ambos procesos se realizan de forma simultánea, aunque en ocasiones no se emplee el resultado de la generación.

Dentro de este capítulo, en el siguiente apartado se explica la arquitectura del sistema y sus módulos fundamentales. Para cada uno de ellos se especifica un pseudocódigo con su funcionalidad. En el apartado 5.3.2 se detalla un entorno de demostración que servirá como modelo para desarrollar cada uno de los pseudocódigos y explicar los procesos de interacción (apartado 5.3.3) y generación (apartado 5.3.4).

5.3.1 Esquema del sistema de interpretación y generación

El sistema de interpretación y generación automáticos sigue el esquema representado en la Figura 40. La arquitectura se compone de cinco módulos fundamentales: el Módulo de Procesamiento del Resultado (MPR), el Módulo de Mapeo del Resultado (MMR), el Módulo de Procesamiento de Nodos (MPN), el Módulo de Procesamiento del Árbol (MPA) y el Módulo de Adición de Palabras (MAP). Además, existen módulos adicionales que permiten reproducir la respuesta de clarificación generada y ejecutar la acción solicitada, así como un historial de las oraciones pronunciadas y las acciones llevadas a cabo. Los cuatro primeros módulos se utilizan tanto durante el proceso de interpretación de oraciones (ver más adelante el apartado 5.3.3) como durante el de generación (ver más adelante el apartado 5.3.4) mientras que el quinto módulo se emplea únicamente para la generación de oraciones.

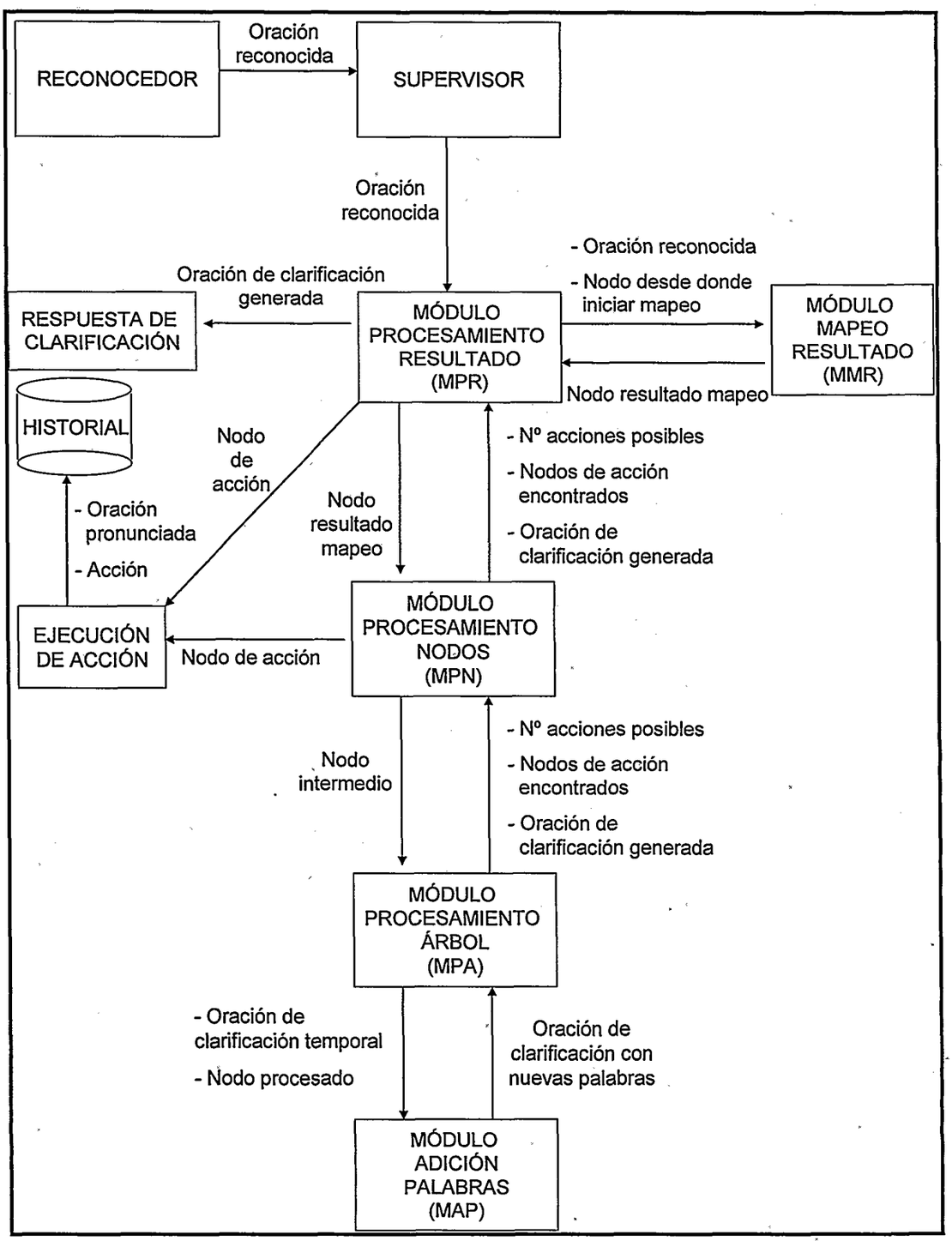


Figura 40. Arquitectura del sistema de interpretación y generación

La funcionalidad básica de cada uno de estos módulos es:

- El MPR es un distribuidor de la información por los diferentes módulos que componene la interfaz.
- El MMR busca coincidencias entre la oración pronunciada por el usuario y los nodos del árbol.
- El MPN se ocupa de realizar una acción (en caso de que se haya interpretado completamente la oración pronunciada por el usuario) o de solicitar la clarificación de la acción requerida (si la oración no se interpreta completamente).
- El MPA se encarga de determinar cuál ha sido la solicitud del usuario y de ofrecer las opciones posibles.
- El MAP se ocupa de construir correctamente las oraciones de respuesta al usuario.

A continuación, se verá una descripción más detallada de cada uno de los módulos, de modo que pueda obtener una visión general del sistema.

5.3.1.1 Módulo de Procesamiento del Resultado

El pseudocódigo que describe el proceso de funcionamiento del MPR es:

```
Recibe la oración reconocida del Supervisor y el árbol de
procesamiento lingüístico.

Si en la última iteración se paró en algún nodo intermedio del
árbol:

   Para cada nodo donde se paró:

      Se envía al MMR la oración, el nodo donde se paró, que se ha de
      comprobar si los nodos están habilitados y que no se pueden
      eliminar partes de la oración.

      Si el MMR devulve alguna coincidencia:

         Se envía al MPN el nodo devuelto por el MMR, que no se pueden
         realizar acciones, que se continúa con un diálogo previo y
         que no se ha saltado ningún nodo hijo.

         Si el número de acciones posibles (NAP) devuelto por el MPN
         es superior a cero:

            Suma al número total de acciones (NTA) el NAP devuelto por
            el MPN.

            Añade la oración de clarificación generada (OCG) por el
            MPN a la oración de clarificación final (OCF).

Si no o si el MPN no devolvió ninguna coincidencia:

   Se envía al MMR la oración, el nodo raíz del árbol lingüístico,
   que no se ha de comprobar si los nodos están habilitados y que se
```

pueden eliminar partes de la oración.

Si el MMR devuelve alguna coincidencia:

Se envía al MPN el nodo devuelto por el MMR, que se pueden realizar acciones, que no se continúa con un diálogo previo y que no se ha saltado ningún nodo hijo.

Si el nodo devuelto por el MPN es un nodo de resolución de la anáfora:

Se llama recursivamente al MPR con la acción referida en la anáfora y la última frase pronunciada.

Si no, suma al NTA las acciones posibles devueltas por el MPN.

Si no o si el MPN no devolvió ninguna coincidencia y si en la última interacción se paró en un nodo intermedio del árbol:

Para cada nodo donde se paró:

Para cada hijo del nodo donde se paró:

Se envía al MMR la oración, el nodo donde se paró, que se ha de comprobar si los nodos están habilitados y que no se pueden eliminar partes de la oración.

Si el MMR devuelve alguna coincidencia y se trata de una acción distinta:

Se envía al MPN el nodo devuelto por el MMR, que no se pueden realizar acciones, que se continúa con un diálogo previo y que se ha saltado algún nodo hijo.

Si el NAP devuelto por el MPN es superior a cero:

Suma al NTA las acciones posibles devueltas por el MPN.

Añade la OCG por el MPN a la OCF.

Si no o si el MPN no devolvió ninguna coincidencia:

Para cada hijo del nodo raíz del árbol lingüístico:

Se envía al MMR la oración, el nodo donde se paró, que no se ha de comprobar si los nodos están habilitados y que no se pueden eliminar partes de la oración.

Si el MMR devuelve alguna coincidencia y se trata de una acción distinta:

Se envía al MPN el nodo devuelto por el MMR, que no se pueden realizar acciones, que no se continúa con un diálogo previo y que ha saltado algún nodo hijo.

Si el NAP devuelto por el MPN es superior a cero:

Suma al NTA las acciones posibles devueltas por el MPN.

Añade la OCG por el MPN a la OCF.

```
Si el NTA es superior a tres:
   Crea una oración de ayuda con todas las opciones posibles a
   partir de la OCF.
   Si la OCF contiene varios verbos:
      Procede a la reducción de la OCF con varios verbos.
   Si no:
      Procede a la reducción de la OCF con un solo verbo.
Se elimina de la oración las palabras para eliminar devueltas por el
MMR.
Si el NTA es igual a uno y todavía no se ha ejecutado la acción
pertinente:
   Ejecuta la acción correspondiente.
   Almacena la oración reconocida original y la acción en el
   Historial.
Si el el MPN no ha devuelto ningún nodo válido:
   Se envía al sintetizador de voz una oración para informar que no
   se ha podido interpretar la oración reconocida.
Si el NTA es superior a uno:
   Si todavía quedan partes de la oración por procesar:
      Se llama de forma recursiva al MPR con la oración con las
      partes procesadas eliminadas.
   Si no:
      Se envía al sintetizador de voz la OCF.
```

Figura 41. Pseudocódigo del Módulo de Procesamiento del Resultado

5.3.1.2 Módulo de Mapeo del Resultado

Su pseudocódigo es el siguiente:

```
Recibe la oración que se desea mapear, el nodo desde donde iniciar
la comprobación, si se ha de comprobar el estado de los nodos del
árbol y si se pueden eliminar partes de la oración.

Hace:

   Para cada nodo hijo del nodo recibido:

      Si se ha de comprobar el estado de los nodos y el nodo hijo
      está habilitado y se produce una coincidencia en la oración con
      el nodo hijo o si no se ha de comprobar el estado de los nodos
      y se produce una coincidencia en la oración con el nodo hijo:

         Marca que se ha producido una coincidencia.

   Si se ha producido alguna coincidencia:

      Asigna como nuevo nodo que inspeccionar el nodo hijo donde se
      produjo la coincidencia.

      Si se pueden eliminar palabras:

         Elimina de la oración la palabra donde se produjo la
         Coincidencia.

      Si no:

         Añade la palabra a la lista de palabras para eliminar que se
         devuelve al MPR.

Mientras que se encuentren coincidencias y la oración contenga
Palabras.

Devuelve al MPR el último nodo donde se produjo una coincidencia y
la lista de palabras para eliminar.
```

Figura 42. Pseudocódigo del Módulo de Mapeo del Resultado

5.3.1.3 Módulo de Procesamiento de Nodos

La descripción de su pseudocódigo es la siguiente:

```
Recibe el nodo que se ha de procesar, si el módulo puede realizar
acciones, si se continúa con un diálogo anterior y si se ha saltado
algún nodo hijo.

Hace:

   Cuenta cuántos hijos activos tiene el nodo recibido.

   Si sólo tiene un hijo activo:

      Asigna como nodo para procesar el único nodo hijo activo.

Mientras sólo haya un único hijo activo.

Si pueden realizar acciones y es un nodo de acción:

   Se ejecuta la acción asociada al nodo.

   Devuelve que el número de acciones posibles es uno.

Si no:

   Envía al Módulo de Procesamiento del Árbol el nodo, si se pueden
   realizar acciones, si se continúa con un diálogo anterior y si se
   ha saltado algún nodo hijo.

Si no se ha ejecutado ninguna acción y el nodo procesado no es un
nodo hoja o si el nodo procesado inicia un subdiálogo:

   Añade este nodo a la lista de nodos procesados que se utilizará
   para continuar con el diálogo en la siguiente interacción.
```

Figura 43. Pseudocódigo del Módulo de Procesamiento de Nodos

5.3.1.4 Módulo de Procesamiento del Árbol

El pseudocódigo con el comportamiento que sigue este módulo es el siguiente:

```
Recibe el nodo desde el que debe procesar el árbol, si es posible
realizar acciones, si se está continuando con un diálogo previo y si
se ha saltado algún nodo hijo.

Si no es un nodo de acción:

   Realiza una búsqueda recursiva a través del árbol.

   Si la oración de clarificación generada durante la búsqueda
   recursiva ofrece varias alternativas sin una o disyuntiva final:

      Sustituye la última , por una o.

   Si se ha saltado un nodo hijo y el número de acciones posibles es
   superior a cero:

      Si el nodo abuelo del nodo recibido no está en la lisa de nodos
```

```
        ya procesados en la actual interacción:

            Añade en la lista de nodos procesados que se utilizarán en la
            próxima iteración el nodo abuelo del recibido.

            Si el nodo de parte verbal correspondiente con el camino del
            árbol hasta llegar al nodo recibido no está en la lista de
            nodos procesados en la actual interacción:

                Añade el nodo de parte verbal a la lista de nodos
                procesados que se utilizará en la próxima interacción.

    Si no, si la acción solicitada por el usuario es diferente al estado
    actual de alguna entidad asociada al nodo:

        Si el nodo abuelo del nodo recibido no está en la lisa de nodos
        ya procesados en la actual interacción:

            Añade en la lista de nodos procesados que se utilizarán en la
            próxima iteración el nodo abuelo del recibido.

            Si el nodo de parte verbal correspondiente con el camino del
            árbol hasta llegar al nodo recibido no está en la lista de
            nodos procesados en la actual interacción:

                Añade el nodo de parte verbal a la lista de nodos procesados
                que se utilizará en la próxima interacción.

    Si se pueden realizar acciones y el número de acciones posibles es
    igual a cero:

        Se genera una oración de clarificación informando de que no hay
        elementos en el entorno que permitan realizar la acción
        solicitada.

    Si no, si se pueden realizar acciones y el número de acciones
    posibles es igual a uno:

        Se realiza la acción pertinente.

        Se guarda en el historial la acción y oración pronunciada.

    Si no, si se pueden realizar acciones y el número de acciones
    posibles es superior a tres:

        Crea una oración de ayuda con todas las opciones posibles a
        partir de la oración de creada durante la búsqueda recursiva.

        Procede a la reducción de la oración creada con un verbo y la
        asigna como oración de clarificación final.

    Si no:

        Asigna a la oración creada en la búsqueda recursiva el estado de
        oración de clarificación final.
```

Figura 44. Pseudocódigo del Módulo de Procesamiento del Árbol

El pseudocódigo de procesamiento de la búsqueda recursiva es el siguiente:

```
Para cada nodo hijo del nodo recibido:

  Si el nodo hijo tiene asociada alguna entidad que no ha sido
  procesado por el resto de los hijos al mismo nivel:

    Si el nodo hijo es un nodo de acción y la entidad del nodo hijo
    tiene un estado distinto al solicitado por el usuario:

      Incrementa en uno el número de acciones posibles.

      Envía la palabra del nodo actual al Módulo de Adición de
      Palabras para que la añada a la suboración de clarificación
      generada hasta ese nivel.

      Añade a la oración de clarificación que devuelve la
      suboración generada hasta ese nivel.

  Si no:

    Envía la palabra de nodo actual al módulo de adición de
    palabras para que la añada a la suboración de clarificación que
    se va generando.

    Llama de forma recursiva a la búsqueda en árbol con el nodo
    Hijo.
```

Figura 45. Pseudocódigo del procesamiento de búsqueda recursiva

5.3.2 Entorno de demostración

Los ejemplos y explicaciones empleadas en la siguiente sección se basarán en un entorno sencillo en donde sólo se encuentran tres entidades: *Fluor, Lamp* y *Radio*. La primera corresponde con una luz fluorescente. La segunda con una lámpara halógena regulable y la tercera con un sintonizador de radio con dos emisoras. Para poder representar gráficamente el árbol lingüístico correspondiente se han limitado considerablemente las posibles interacciones, de modo que la información lingüística asociada a cada entidad (ver apartado 5.1.1) es la que aparece representada en la Figura 46, Figura 47 y Figura 48.

```
<class name="fluor">

   <property name="status">
      <paramSet name="dialogue">
         <param name="action1">encender</param>
         <param name="skeleton1">VP encender OP luz MP fluorescente
         </param>
      </paramSet>
   </property>

</class>
```

Figura 46. DPCE con información lingüística de la entidad Fluor

```
<class name="lamp">

   <property name="status">
      <paramSet name="dialogue">
         <param name="action1">encender</param>
         <param name="skeleton1"> VP encender OP luz lámpara
         MP halógena </param>
      </paramSet>
   </property>

   <property name="value">
      <paramSet name="dialogue">
         <param name="action1">subir</param>
         <param name="skeleton1"> VP subir OP luz lámpara MP halógena
         </param>
      </paramSet>
   </property>

</class>
```

Figura 47. DPCE con información lingüística de la entidad lamp

```
<class name="radio">

   <property name="status">
      <paramSet name="dialogue">
         <param name="action1">encender_m_80</param>
         <param name="skeleton1"> VP encender poner OP radio
         SUB MP m_80 </param>
         <param name="action2">encender_radio_5</param>
         <param name="skeleton2"> VP encender poner OP radio
         SUB MP radio_5 </param>
      </paramSet>
   </property>

   <property name="value">
      <paramSet name="dialogue">
         <param name="action1">subir</param>
         <param name="skeleton1"> VP subir OP volumen MP radio
</param>
         <param name="skeleton2> VP subir OP radio </param>
      </paramSet>
   </property>

</class>
```

Figura 48. DPCE con información lingüística de la entidad Radio

La entidad Fluor sólo tiene una posible acción que se invoca mediante interacciones del tipo *encender luz flourescente*, producidas por frases como *Quiero que enciendas la*

luz del fluorescente o *Podrías encender la luz del fluorescente*. Además, en este caso no es necesario proporcionar toda la información. La interacción *Quiero que enciendas el fluorescente* es igualmente válida, ver más adelante el apartado 5.3.3.6.

La entidad Lamp tiene dos posibles acciones *encender* y *subir*. En este caso *luz* y *lámpara* se presentan como sinónimos por lo que resulta equivalente decir *Me gustaría que se encendiera la luz halógena* o *Me gustaría que se encendiera la lámpara halógena*. Eso no ocurría en la entidad *Fluor* donde la interacción *Quiero que enciendas la lámpara fluorescente* se considera como no válida.

La entidad *Radio* presenta tres acciones: *encender la emisora m_80*, *encender la emisora radio_5* y *subir el volumen*. Las dos primeras presentan un subdiálogo tras la parte *radio* que inicia la solicitud de la emisora que se desea escuchar. La acción *subir* tiene dos posibilidades de interacción equivalentes: *subir volumen radio* y *subir radio*.

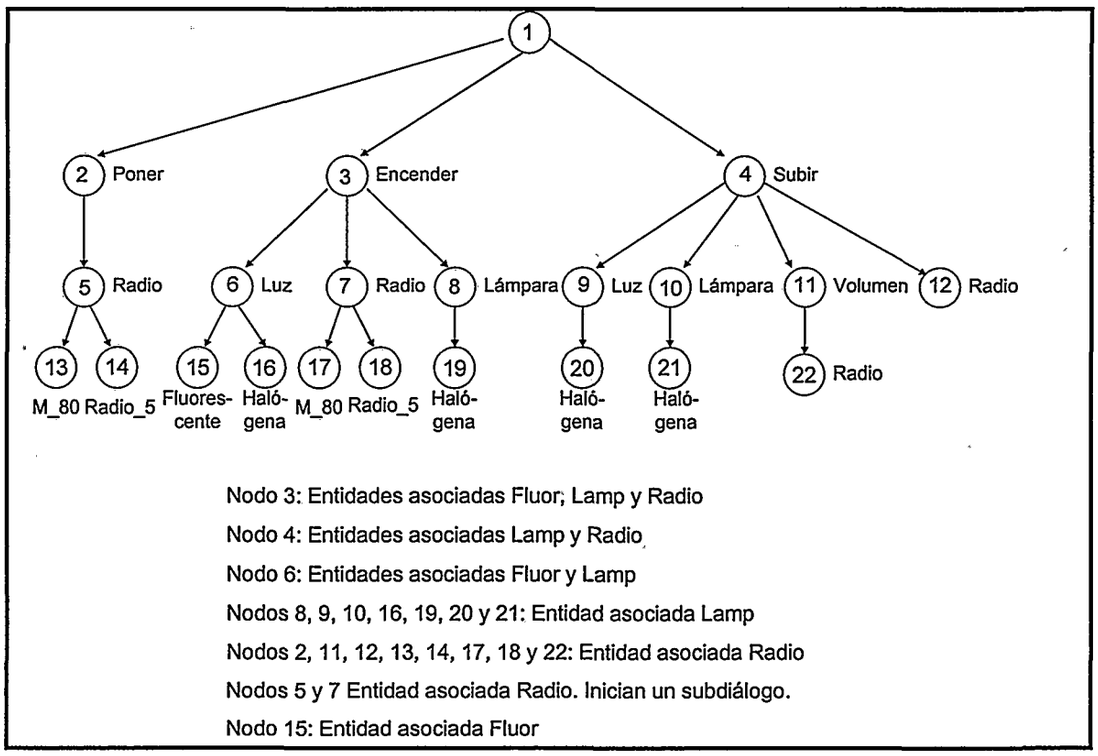


Figura 49. Árbol lingüístico para las entidades Fluor, Lamp y Radio

Con esta información lingüística se crea automáticamente el árbol representado en la Figura 49 que servirá de base para la interpretación y generación de oraciones que debe realizar el supervisor.

5.3.3 Interpretación de oraciones

Cuando el supervisor recibe una oración del reconocedor la envía al Módulo de Procesamiento del Resultado (MPR), que consiste en un distribuidor de la información por los diferentes módulos que componen la interfaz.

Desde aquí la información se envía al Módulo de Mapeo del Resultado (MMR) que recibe un nodo que actúa como raíz del árbol lingüístico sobre el que se realizará el mapeo y la oración reconocida. Este módulo comprueba si se producen coincidencias de la oración con los nodos del árbol recibido, descendiendo un nivel en el mismo por cada coincidencia.

Por ejemplo, basándose en el árbol representado en la Figura 49, si tras despertar al sistema (ver apartado 3.9.2.3), el usuario pronuncia *Me gustaría que encendieras la lámpara* el reconocedor envía al MPR la oración *encender lámpara* (ver apartado 3.9.2.2). Este, a su vez, envía esta oración al MMR que procesa la oración empezando desde la raíz del árbol lingüístico. La palabra *encender* coincide con el nodo número 3, así que desciende a ese nodo y sigue buscando coincidencias. La palabra *lámpara* concuerda con el nodo número 8 así que desciende a ese nodo. A partir de este punto no se producen más coincidencias, por lo que el MMR devuelve al MPR el nodo obtenido como resultado de la búsqueda en el árbol (esto es, el nodo *lámpara* representado en la figura con el número 8).

Si el MPR recibe un nodo de respuesta es porque se ha producido alguna coincidencia con el árbol lingüístico. Esto significa que el sistema está en condiciones de interpretar lo que el usuario ha pronunciado, por lo que envía la información del nodo al Módulo de Procesamiento de Nodos (MPN).

El MPN debe determinar si ha interpretado completamente la oración pronunciada por el usuario (y por lo tanto puede ejecutar la acción requerida) o si todavía necesita conocer más información, por lo que debe realizar un proceso de clarificación.

5.3.3.1 Ejecución de una acción en el entorno

Si el nodo recibido por el MPN es un *nodo de acción* el módulo ha interpretado completamente el requerimiento del usuario y está en condiciones de atenderlo ejecutando la acción pertinente en el entorno.

Un *nodo de acción* corresponde con un nodo hoja o con un nodo que inicia un subdiálogo:

- Los nodos hoja sólo pueden tener una única entidad y una acción en la lista de entidades y acciones asociadas al nodo. Se alcanzan estos después de seguir un camino que identifica de forma unívoca la acción que se requiere realizar con una entidad.

- Los nodos que inician un subdiálogo corresponden con aquellos que tienen alguna entidad con esta propiedad. Dentro de la lista de entidades y acciones asociadas al nodo, la entidad y acción que se utilizarán para la ejecución corresponden con las que fueron definidas como evocadoras del subdiálogo (ver apartado 5.1.1.1).

En cualquiera de los dos casos se llama de forma automática al método *BBAction* de la entidad correspondiente. Como se definió en el apartado 5.1.3.1 este método recibe la entidad de la pizarra sobre la que se realiza la acción, la oración pronunciada por el usuario y la acción requerida. Con estos datos el método, en el caso de ser posible, realiza una acción en el entorno físico comunicándose mediante mensajes con la pizarra (ver apartado 3.4).

Aunque la implementación de este método corre por cuenta del diseñador de la interfaz de comunicación con el tipo de entidad al que pertenece, su proceso de ejecución suele seguir unos pasos sencillos:

- Se comprueba el estado físico actual de la entidad (leyendo la información de la pizarra) y se contrasta con el solicitado por el usuario.
- En el caso de que ambos sean iguales se informa al usuario de que la entidad ya se encuentra en el estado solicitado.
- En caso contrario se lleva a cabo la acción requerida enviando las órdenes estándar a la entidad de la pizarra.
- Si el nodo que invocó la acción correspondía con un subdiálogo el método no realizará ninguna acción física en el entorno, sino que solicitará o planteará al usuario nueva información específica que permita continuar con el diálogo.

Tras llamar al método y ejecutar la acción pertinente el MPN almacena el nodo en un histórico de las acciones realizadas y anota que el número de acciones que ha interpretado como posibles en el entorno en la presente interacción es igual a uno.

5.3.3.2 Clarificación de la oración

Si el MPN no ha recibido un *nodo de acción* resulta necesario clarificar la solicitud realizada por el usuario. Para esto envía el nodo al Módulo de Procesamiento del Árbol (MPA) que se encarga de determinar qué ha querido solicitar el usuario y, en su caso, ofrecerle las opciones que tiene disponibles.

El MPA visita los hijos del nodo recibido, determina cuántas opciones se pueden realizar de acuerdo con el contexto físico del entorno representado en la pizarra (ver apartado 3.3) y genera una oración de respuesta adecuada. El recorrido de los nodos se basa en una búsqueda recursiva del árbol en profundidad. Su procedimiento consiste en:

- Visita al primer hijo del nodo recibido y comprueba si el nodo está asociado a alguna entidad que todavía no haya sido procesada a ese nivel. Si no es así, el nodo

hijo corresponde con un sinónimo de algún nodo previamente procesado al mismo nivel, por lo que se ignora y se pasa al siguiente hijo.

- Si se puede procesar añade las entidades que tiene asociadas a la lista de entidades ya procesadas en ese nivel y comprueba si este nodo hijo corresponde con un *nodo de acción*.
- Si es un *nodo de acción* se verifica si el estado de la entidad asociada al nodo hijo es distinto al estado solicitado por el usuario. Para ello se llama de forma automática al método *hasRequestedADifferentAction* de la entidad (ver apartado 5.1.3.2). Este método recibe la entidad de la pizarra asociada al nodo, la oración pronunciada y la acción solicitada por el usuario.
- En el caso de que el nodo presente un estado distinto al requerido por el usuario se considera a este nodo como un *nodo de acción válido*, incrementa en uno el número de opciones que se pueden ofrecer al usuario y guarda el nodo en la lista de *nodos de acción válidos* para la actual solicitud del usuario.
- Si el nodo hijo no es un *nodo de acción* se continúa de forma recursiva con los pasos anteriormente descritos, procesando el primer hijo del nodo hijo actual.
- Este proceso se sigue con cada uno de los hijos de cada nodo hasta que se han procesado todos los nodos pertinentes por debajo del nodo recibido. Por nodo pertinente se considera aquel que ofrece un estado distinto al solicitado por el usuario y que no corresponde con un sinónimo de otro ya procesado.

Como resultado de esta búsqueda en el árbol, se obtiene el número de entidades del entorno que tienen un estado distinto al solicitado por el usuario (esto es, el número de acciones que se pueden ofrecer), el listado de los *nodos de acción* que representan estas acciones y una oración de clarificación que se ha generado automáticamente durante el proceso de búsqueda (ver más adelante el apartado 5.3.4). Esta información se devuelve al MPN que termina añadiendo el nodo a la lista de nodos hasta donde ha sido posible procesar la información con la actual oración (esto es, a partir de donde se ha realizado la clarificación).

El MPR vuelve a tomar el control y dependiendo del número de acciones que se pueden ofrecer al usuario se comporta de una de las siguientes maneras:

- Si el número de acciones es igual a cero significa que no existe en ese momento ninguna entidad con un estado distinto al requerido por el usuario. Empleando el sintetizador de voz se informa sobre este hecho.
- Si el número de acciones es igual a uno, el MPR ejecuta la acción correspondiente utilizando el método *BBAction* de la entidad asociada al único *nodo de acción* de la lista de *nodos de acción* devuelta (ver apartado anterior). De esta forma se reducen el número de turnos, evitando realizar una pregunta de clarificación. Además, gracias al uso de la información del contexto físico almacenada en la pizarra, el

sistema se ha podido recuperar de un error del reconocedor que ha devuelto menos información de la expresada por el usuario o ha interpretado de forma correcta una oración que no contenía la información completa necesaria.

- Si el número de acciones es superior a uno el MPR pronuncia la oración que se ha generado durante el proceso de búsqueda en el árbol o una nueva oración similar (ver más adelante el apartado 5.3.4). Esta oración informa sobre las posibles acciones que se pueden llevar a cabo en el entorno según la información suministrada por el usuario y el contexto del mismo. De este modo se asiste y guía a la persona que utiliza la interfaz en el proceso de interacción con el entorno.

5.3.3.3 Variaciones en la interpretación dependiendo del contexto

Como se ha descrito en el apartado anterior la interpretación depende de forma sustancial del contexto actual del entorno. Las oraciones pueden conducir a interpretaciones muy diferentes en contextos distintos. Como ejemplo, en el entorno mostrado en la Figura 49 supongamos dos casos diferentes que comparten el mismo escenario: la entidad *Fluor* está encendida y las entidades *Radio* y *Fluor* están apagadas.

En el primer escenario el usuario pronuncia la oración *Enciende, por favor.* El proceso se desarrollaría de la siguiente manera:

- El MPR recibe la oración *encender* del reconocedor y la envía al MMR indicándole que empiece a buscar desde la raíz del árbol lingüístico.
- Este obtiene coincidencias en el árbol hasta el nodo *Encender* (el número 3 de la figura) por lo que devuelve este nodo al MPR.
- El MPR envía el nodo al MPN para que determine cómo interpretar la oración. Dado que el nodo recibido no corresponde con un *nodo de acción* (el nodo 3 no es un nodo hoja ni inicia ningún subdiálogo) determina que se necesita clarificar la oración reconocida.
- El MPN envía el nodo al MPA para que busque en sus hijos las posibles opciones.
- En primer lugar el MPA desciende al nodo *Luz* (el nodo 6). Como a su vez este nodo tiene dos hijos continúa con el primero de ambos, esto es, con el nodo *Fluorescente* (el nodo número 15).
- Comprueba que este nodo no corresponde con ningún sinónimo de alguno ya procesado a este nivel. Al ser un *nodo de acción,* mediante el método *hasRequestedADifferentAction* de la entidad *Fluor* (la que está asociada a este nodo), verifica que el estado de la entidad (encendido) es el mismo al que ha solicitado el usuario (encender). Por lo tanto no considera este nodo como un *nodo de acción válido.*

- A continuación el MPA desciende al nodo *Halógena* (nodo número 16). Tras comprobar qué no se trata de ningún sinónimo, que es un *nodo de acción* y que el estado de su entidad asociada (*Lamp*) es distinto al solicitado por el usuario incrementa a uno el número de acciones posibles y añade el nodo a la lista de *nodos de acción válidos*.
- El MPA continúa la búsqueda por el nodo *Radio* (el número 7). Comprueba que no es un sinónimo y, al ser un nodo que inicia un subdiálogo, se considera como un *nodo de acción*. Verifica que su estado (apagado) es distinto al solicitado (encender) por lo que se incrementa a dos el número de acciones posibles y se añade a la lista de *nodos de acción válidos*.
- El MPA sigue la búsqueda con el nodo *Lámpara* (el nodo número 8). Este nodo corresponde con un sinónimo de uno anteriormente procesado (su única entidad asociada, *Lamp,* ya fue anteriormente procesada por el nodo 6) por lo que no se considera.
- El MPA ha terminado la búsqueda en el árbol y envía la información al MPN.
- Este añade el nodo *Encender* (el número 3) a la lista de nodos hasta donde se ha procesado la información con la oración actual.
- El MPR toma el control y ve que el número de acciones posibles es igual a dos. Con los datos obtenidos pronuncia la oración que se ha generado automáticamente *Prefieres encender la luz halógena o la radio* (ver más adelante el apartado 5.3.4).

En el segundo escenario el usuario pronuncia la oración *Me gustaría que encendieras la luz*. En este caso la interpretación varía de la siguiente manera:

- El MPR recibe la oración *encender luz* del reconocedor y la envía al MMR indicándole que empiece a buscar desde la raíz del árbol lingüístico.
- El MPR recibe como resultado la coincidencia con el árbol lingüístico del nodo *Luz* (el nodo 6).
- El MPN y el MPA actúan de forma similar a la expuesta en el escenario anterior. En este caso se obtiene que el número de acciones posibles es uno y que el único *nodo de acción* válido es el nodo *Halógena* (el número 16).
- El MPR, al obtener que sólo existe una posibilidad, ejecuta directamente la única acción posible, esto es, enciende la luz halógena. Con esto, gracias a la ayuda del contexto, el sistema ha interpretado qué acción quería realizar el usuario a pesar de que éste ofreció menos información de la necesaria.

Estas mismas oraciones pueden tener resultados distintos en contextos diferentes. Supongamos ahora un nuevo escenario donde la entidad *Radio* está apagada y las entidades *Lamp* y *Fluor* están encendidas:

- Si el usuario vuelve a pronunciar *Enciende, por favor* el sistema ahora encenderá la radio, ya que ésta es la única entidad que se puede encender y que en esos momentos se encuentra apagada.
- Si el usuario pronuncia ahora nuevamente la oración *Me gustaría que encendieras la luz*, el sistema obtiene que no hay ninguna entidad que puede cumplir la acción solicitada por lo que informa al usuario que todas las luces del entorno ya se encuentran encendidas.

Como se ha visto en estos ejemplos el proceso de interpretación se adapta de forma automática las entidades presentes en el entorno y a su contexto. El entorno puede disponer de más o menos entidades, éstas pueden cambiar su estado físico o puede haber múltiples entidades del mismo tipo. La interfaz se adecua a estas circunstancias alterando automáticamente su comportamiento.

5.3.3.4 Continuación de un diálogo previo

Tras una pregunta de clarificación, el sistema da prioridad a la continuación con el diálogo iniciado anteriormente, aunque también permite empezar un diálogo nuevo (recuperándose así de una posible clarificación errónea en el proceso de interpretación).

Para realizar esta tarea, cuando se formula la respuesta de clarificación el MPR habilita todos los nodos que establecen un camino desde el nodo donde se ha iniciado la búsqueda en el árbol hasta los *nodos de acción válidos*, quedando los demás deshabilitados.

Tras la siguiente oración que pronuncie el usuario el MMR no empieza a buscar coincidencias en el árbol lingüístico desde su raíz, sino que realiza la búsqueda desde el nodo donde se paró en la interacción anterior. Además, el MPA sólo tendrá en cuenta aquellos nodos que se encuentren habilitados, por entender que son los únicos que corresponden a las acciones que se han ofrecido anteriormente al usuario.

Sólo en el caso de que no se produzca ninguna coincidencia tras este análisis, el MMR inicia una nueva búsqueda empezando desde la raíz de árbol lingüístico, tal y como se ha explicado en el apartado 5.3.3.2. Además, en ese caso, el MPA considerará que todos los nodos se encuentran habilitados, ya que ahora no se está continuando con ningún diálogo previo.

De esta forma se permite continuar con una interacción iniciada por el usuario que todavía no ha concluido, aunque también es posible iniciar un nuevo diálogo dejando en suspenso el anterior.

Veamos cómo se produce este proceso basándonos en el primero de los escenarios propuestos en el apartado anterior: la entidad *Fluor* está encendida y las entidades *Radio* y *Lamp* están apagadas. El usuario ha pronunciado *Enciende, por favor*, el sistema ha obtenido que los *nodos de acción válidos* son *Radio* (el número 7) y *Halógena* (el número 16), ha guardado el nodo *Encender* (el número 3) como el nodo

procesado en la presente interacción y ha emitido la oración *¿Prefieres encender la luz halógena o la radio?*.

Ahora, antes de pasar a la siguiente interacción, el sistema habilita todos los nodos que se encuentran desde el nodo *Encender* hasta los nodos *Radio* y *Halógena*, esto es, habilita los nodos *Luz* (el número 6), *Radio* (el número 7) y *Halógena* (el número 16).

Si en la siguiente interacción el usuario pronuncia *La luz* el MMR empieza a comprobar coincidencias en el árbol desde el nodo *Encender* (el número 3) por lo que se obtiene una coincidencia con el nodo *Luz* (el número 6). En este caso el MPA sólo tendrá en cuenta para la búsqueda recursiva de los hijos a aquellos que se encuentren habilitados. De este modo se desestima directamente al nodo *Fluorescente* (el número 15) y sólo se considera al nodo *Halógena* (el número 16). Como el número de acciones posibles es igual a uno el sistema ejecuta directamente la acción, esto es, enciende la luz halógena.

De esta forma no sólo hemos visto cómo el sistema permite continuar con un diálogo que se inició previamente sino que se ha comprobado un nuevo ejemplo donde se interpreta correctamente la oración pronunciada por el usuario a pesar de que este proporcionó menos información de la precisa (aunque el usuario sólo se ha referido a la luz sin especificar cuál de las dos, el sistema interpreta que se refiera a la halógena ya que anteriormente la luz fluorescente no fue ofrecida en la lista de acciones posibles).

5.3.3.5 Interpretación inicial de oraciones incompletas

Como se ha visto en los apartados anteriores la interfaz es capaz de recuperarse de diversos errores, entre ellos de la recepción de oraciones incompletas. Este suceso se puede producir bien porque el usuario no facilite toda la información necesaria para determinar la acción que se debe llevar a cabo o bien porque el reconocedor de voz no haya sido capaz de devolver la oración completa pronunciada por el usuario.

Además de los mecanismos ya vistos, la interfaz incorpora un modo de interpretación avanzado que intenta descifrar oraciones que sólo proporcionan parte de la información. Este mecanismo consiste en explorar el árbol a partir de los hijos de los nodos donde no se producen coincidencias, considerando que la información referida a estos nodos ha podido ser omitida.

En estos casos la búsqueda de coincidencias en el árbol lingüístico sigue el siguiente orden:

- Primero el MMR busca coincidencias desde el nodo donde se paró en la anterior interacción.
- En el caso de que no exista este nodo o de que no se haya encontrado ninguna coincidencia, el MMR empieza buscando desde la raíz del árbol.

- Si tampoco se producen aquí coincidencias el MMR realiza las comprobaciones empezando en cada uno de los hijos del nodo donde se paró en la anterior interacción, esto es, se salta la comprobación para cada uno de estos nodos hijo.
- Si finalmente tampoco se producen coincidencias el MMR realiza las comprobaciones empezando en cada uno de los hijos del nodo raíz.

Para realizar esta búsqueda con salto de nodos se debe tener en cuenta además que sólo se han de considerar los hijos que no correspondan con sinónimos de nodos ya procesados.

Cada uno de los nodos hijo a los que se ha saltado proporcionará un conjunto completo de interacciones posibles. Al final, el total de interacciones posibles vendrá dado por la suma de las interacciones posibles de cada conjunto.

Para ver cómo funciona este mecanismo supongamos un escenario donde la entidad *Lamp* está encendida y la entidad *Fluor* está apagada. El usuario pronuncia *Me gustaría que encendieras la luz* pero el reconocedor sólo es capaz de devolver la palabra *Luz*. El proceso de buscar coincidencias en el árbol lingüístico sigue los siguientes pasos:

- El MMR intenta buscar coincidencias empezando desde el nodo donde se paró en la anterior interacción. Como se inicia una nueva interacción este nodo no existe así que pasa al siguiente caso.
- El MMR busca coincidencias empezando desde la raíz del árbol lingüístico. *Luz* no concuerda con *Poner* (nodo número 2), *Encender* (nodo número 3) y *Subir* (nodo número 4) así que continúa con el caso siguiente.
- El MMR tampoco puede buscar coincidencias empezando con los hijos del nodo donde se paró la última vez así que pasa al último de los casos.
- Finalmente el MMR busca coincidencias a partir de cada uno de los hijos del nodo raíz. Como se ha descrito anteriormente, esto se realiza considerando a cada uno de los hijos desde los que se empieza de forma individual, realizando una búsqueda independiente y completa para cada uno de ellos.

En este último caso el sistema sigue el siguiente proceso:

- Se empieza buscando coincidencias desde el nodo *Poner* (el número 2). Éste sólo tiene un hijo, *Radio* (nodo número 5), que no concuerda con la salida del reconocedor, *Luz*, por lo que no permite realizar ninguna acción y no se tiene en consideración.
- A continuación el proceso sigue con el nodo *Encender* (el número 3). El procesamiento de este nodo devolvería que sólo existe una acción posible que correspondería con el nodo *Fluorescente* (el número 15) ya que es el único que presenta un estado distinto. El camino hasta esto nodo se habilita y se añaden los nodos raíz (el número 1) y *Luz* (el número 6) a la lista de nodos procesados en la presente interacción.

- Por último se continúa con el nodo *Subir* (el número 4). Como resultado se obtiene que existe una posible acción correspondiente con el nodo *Halógena* (el número 20). Nuevamente se habilita la ruta hasta este último nodo y se añade el nodo *Luz* (el número 9) a la lista de nodos procesados en esta interacción.

Como resultado final el sistema ha obtenido que:

- Existen dos posibles acciones
- Los nodos procesados en la presente interacción son los números 1, 6 y el 9.
- Se han generado dos oraciones de clarificación distintas (*encender luz fluorescente* y *subir luz halógena*). La interfaz combina ambas oraciones para crear una respuesta de clarificación única: *Prefieres encender la luz del fluorescente o subir la luz halógena* (ver más adelante el apartado 5.3.4.4).

Si a continuación el usuario simplemente pronunciara *Prefiero subir* el MMR recibe *subir*. Entonces éste empieza buscando coincidencias con algún nodo del diálogo anterior. En este caso se producirá con el nodo raíz (el número 1). Al producirse una coincidencia con un nodo anterior sólo se procesarán aquellos nodos del árbol que estén habilitados por lo que se llegará al nodo *Halógena* (el número 20). El sistema directamente interpretará que el usuario quiere subir la intensidad de la luz halógena (al provenir de un diálogo previo se desestima la posibilidad de que el usuario quisiera subir el volumen de la radio) y realiza esta acción.

Si en su lugar, el usuario hubiera pronunciado *Prefiero la luz del fluorescente* el MMR encontraría una coincidencia con el nodo *Luz* (el número 6) del diálogo previo, por lo que interpretaría que la oración completa era *encender luz fluorescente* y realizaría esta acción.

5.3.3.6 Interpretación avanzada de oraciones incompletas

Una vez realizada la interpretación completa de una oración y antes de ofrecer las posibles opciones válidas al usuario se llama de forma recursiva al MPR con las partes de la oración pronunciada por el usuario que todavía no han sido procesadas. Sin embargo el MMR sólo buscará coincidencias con los hijos del nodo procesado antes de la llamada recursiva. Esto es, busca partes de información que hayan podido ser omitidas en medio de la oración. Este proceso recursivo se continuará hasta que no queden más palabras en la oración por procesar o hasta que no se produzcan nuevas coincidencias. De este modo el sistema es capaz de recuperarse e interpretar oraciones donde incluso faltan múltiples partes por especificar.

Supongamos un escenario en donde las tres entidades están apagadas y el usuario pronuncia *Quiero que enciendas el fluorescente*. El MPR envía al MMR *encender fluorescente* y éste encuentra coincidencias hasta el nodo *Encender* (el número 3). Como resultado del proceso de interpretación a partir de ese nodo se obtienen tres posibles acciones válidas, la oración de clarificación *Prefieres encender la luz*

fluorescente, la halógena o la radio y el nodo *Encender* en la lista de nodos procesados. Antes de formular la oración de clarificación al usuario el MPR se vuelve a invocar con los datos del diálogo que se acaban de procesar y la palabra *fluorescente* (la única de la oración pronunciada por el usuario que todavía no ha sido utilizada). En esta llamada recursiva el MMR sólo buscará coincidencias a partir de los hijos del nodo hasta donde se ha llegado anteriormente, esto es, del nodo *Encender*. Con esto, se buscan posibles saltos (partes omitidas) en la oración pronunciada por el usuario. Empezaría por el nodo *Luz* (el número 6) y ahí encontraría una coincidencia con el nodo *Fluorescente* (el número 15). No hallaría coincidencias con el nodo *Radio* (el número 7) y no procesaría el nodo *Lámpara* (el número 8) al tratarse de un sinónimo del nodo *Luz*. Como no quedan más palabras que procesar se terminan las llamadas recursivas, dando por concluido el proceso de interpretación. En este caso se ha avanzado con respecto a la interpretación inicial por lo que se considera que el nuevo nodo alcanzado es *Fluorescente* (el número 15). De este modo se ejecuta la acción de la entidad asociada, esto es, se enciende la luz del fluorescente. Así, se ha conseguido interpretar correctamente una oración donde existían partes omitidas.

5.3.3.7 Interpretación en subdiálogos

Como se ha explicado anteriormente (ver apartado 5.1.1.1) en la definición de las posibles interacciones con una entidad se puede especificar la aparición de subdiálogos. Al alcanzar el punto de inicio de un subdiálogo se produce una acción (normalmente se informa o solicita información al usuario a través de una oración sintetizada) y se permite continuar con alguno de los posibles caminos del subdiálogo.

En la Figura 49 aparecen dos nodos que definen el inicio de subdiálogos. Se trata de los nodos 5 y 7 (ambos con al etiqueta *Radio*). Cuando se alcanza uno de estos nodos se ejecuta automáticamente la acción que determina su entidad asociada pero, al contrario de lo que ocurre cuando se ejecuta una acción en un nodo hoja, no se considera que la interacción haya terminado. En este caso se almacena este nodo en la lista de nodos procesados en la última interacción, de modo que la siguiente oración del usuario permite continuar con el subdiálogo.

Supongamos que el usuario pronuncia *Me gustaría que pusieras la radio*. El MMR encuentra coincidencias hasta el nodo *Radio* (el número 5) y lo envía al MPA. Este comprueba que es un nodo que inicia un subdiálogo por lo que ejecuta la acción correspondiente a la entidad que tiene asociada. En este caso el método *BBAction* (ver apartado 5.1.3.1) de la entidad *Radio* sintetizaría la oración *Qué emisora prefieres poner*. A partir de ahí la interacción seguiría los pasos que se realizaban en la continuación de un diálogo previo (ver apartado 5.3.3.4), permitiendo elegir una emisora de radio (por ejemplo *Me gustaría m_80 radio*) o iniciar un nuevo diálogo.

Como ventaja, los subdiálogos no tienen que recibir la información de forma escalonada e independiente para cada uno de ellos. También es posible proporcionar la información de varios subdiálogos a la vez. Por ejemplo, la interacción anterior sería

equivalente a la solicitud *Podrías poner la radio con m_80* o, tal y como se ha visto en el apartado anterior a *Podrías poner m_80.*

5.3.3.8 Resolución de la anáfora pronominal

El sistema de diálogos incorpora un mecanismo que permite la resolución automática de la anáfora pronominal. Como se especificó en el apartado 5.2.1 por cada verbo que forma parte de la interacción con la entidad se forma su parte pronominal. Esto significa que además de los nodos que hemos visto se añaden al árbol (ver apartado 5.2.2) el sistema adjunta automáticamente un conjunto de nodos de resolución de la anáfora.

Basándonos en el árbol de la Figura 49, además de los nodos representados, de la raíz colgarían lo nodos de resolución de la anáfora *Ponerlo, Encenderlo* y *Subirlo.*

Si el MPA alcanza uno de estos nodos de resolución de la anáfora toma la última acción realizada del histórico de acciones almacenado en la interfaz (ver apartado 5.3.3.1). Entonces vuelve a solicitar la ejecución de la acción pero cambiando el verbo y acción anteriores por los que se especifican en el nodo de resolución de la anáfora.

Supongamos que el usuario pronuncia la oración *Puedes encender la luz halógena.* Como resultado el sistema ejecuta la acción correspondiente al nodo *Halógena* (el número 16) y añade al histórico el nodo *Halógena* como el último que ha realizado una acción. Si a continuación el usuario pronuncia *Súbela, por favor* el MPA encontraría coincidencia con el nodo de resolución de la anáfora *Encenderlo.* Llegado a este punto, el sistema tomaría la última acción del histórico (esto es, la que corresponde con el nodo *Halógena*) y volvería a invocarla considerando que en este caso la acción es *Subir.* Por lo tanto la acción que se realizaría con la entidad del nodo 16 sería *subir luz halógena.* A continuación el usuario podría pronunciar *No, quiero apagarla* y el mecanismo sería similar, provocando esta vez que se apague la luz halógena.

De esta forma se consigue resolver en muchos casos el problema de la anáfora pronominal, que es ampliamente utilizada en el habla espontánea.

5.3.3.9 Resolución de palabras especiales

Dentro del proceso de interacción existen algunas palabras especiales que tienen una significación específica que no se ciñe a entidades particulares del entorno. Ejemplos de estas palabras serían *Todos, Ninguna, Otra vez,* etc. que se podrían utilizar para solicitar que se realicen todas las acciones posibles, que no se realice ninguna o que se repita una acción. La versión actual de la interfaz no es capaz de reaccionar ante estas palabras pero sí que puede hacerlo ante dos casos especiales: *Más* y *Menos.*

- Para la primera de ellas se considera que el usuario quiere repetir la última de las acciones. Si éste ha pronunciado *Por favor, sube la luz* la interfaz subirá la intensidad de la luz halógena. Si a continuación pronuncia *Un poco más* el sistema vuelve a repetir la última acción del histórico, esto es, *subir luz halógena* por lo que

volverá a subirla. Si por el contrario un usuario hubiera pronunciado *Baja la luz halógena* y a continuación *Más, por favor* el sistema bajaría dos veces la intensidad de la luz de la entidad *Lamp*.

- Para la segunda el sistema considera que siempre se desea rebajar la última acción realizada. Por lo tanto si tras la oración *Puedes subir la luz* el usuario pronunciara *Menos, por favor* la interfaz invocaría la última acción del entorno pero considerando que la acción requerida es *Bajar*. En este caso se realizaría la acción *Bajar luz halógena* por lo que se disminuiría la intensidad de la luz.

Para permitir el uso de estas palabras especiales las gramáticas cuentan con unas reglas fijas. En estas reglas sólo se representan entre llaves las palabras *más* y *menos* que son las que el reconocedor devuelve al MPR (ver apartado 3.9.2.2), aunque estas se combinan con otras para permitir que estas instrucciones se empleen de forma más flexible.

5.3.4 Generación de oraciones

La interfaz de diálogos también utiliza el árbol lingüístico para la generación automática de oraciones de clarificación. Como se ha mencionado en el apartado 5.3.3.2 el proceso de generación de la respuesta se realiza a la vez que la clarificación en el proceso de interpretación. Por lo tanto, siempre que la interfaz tiene que realizar una clarificación de la oración pronunciada por el usuario se genera simultáneamente una oración de respuesta que posteriormente podrá ser utilizada o no (se puede dar el caso de que se encuentre una única acción posible y se ejecute directamente).

Esta oración de respuesta se emplea para informar sobre todas las acciones válidas en el entorno, teniendo en cuenta la solicitud realizada por el usuario y el contexto físico del mismo. Con esto se pretende orientar al usuario durante el proceso de interacción con la interfaz, ofreciéndole las posibilidades existentes.

Inicialmente se parte de una oración vacía a la que se van añadiendo palabras utilizando el módulo de adición de palabras (MAP).

5.3.4.1 Módulo de adición de palabras

Este módulo recibe el nodo que contiene la palabra que se quiere añadir y la frase construida hasta el momento. El MAP obtiene del nodo la palabra que contiene (ver apartado 5.2.2), el tipo de parte con que fue definido (ver apartado 5.1.1) y la información del análisis sintáctico que se realizó durante la creación de las gramáticas (ver apartado 5.2.1). Estos datos se utilizan para añadir la nueva palabra a la oración construida hasta el momento. Esta palabra irá precedida, en el caso de ser necesario, de las preposiciones y artículos adecuados.

Para las preposiciones se siguen las siguientes reglas:

- En el caso de que la palabra corresponda con un nombre y se trate de una parte de ubicación o una parte modificadora se antepone la preposición *de*.
- Si la palabra proviene de una parte cuantificadora o de una parte de objeto indirecto se sitúa en un principio la preposición *a*.
- Si la palabra corresponde con una parte modal y no es un adverbio se antepone la preposición *por*.

A continuación se tiene en cuenta la posible adición de un artículo determinado. Este se añadirá en el caso de que la palabra sea un nombre común y corresponda con una parte objeto o una parte de ubicación. Para escoger el artículo correcto se tiene en cuenta si la palabra es masculina o femenina y singular o plural. Además, en el caso de masculino singular se comprobará si anteriormente se añadió la preposición *de* para utilizar la contracción *del*.

Este módulo se llama cada vez que se quiere añadir una nueva palabra a la oración que se va creando durante el proceso de generación automática de la respuesta.

5.3.4.2 Proceso de generación

El proceso de generación se realiza en el MPA junto con algunos de los pasos realizados durante la clarificación (ver apartado 5.3.3.2).

Las palabras se añaden utilizando el MAP descrito en el apartado anterior en diferentes suboraciones. Cada suboración está relacionada con el camino existente desde el nodo donde se inicia la clarificación hasta cada hoja.

Cada vez que el MPA accede a un nodo del árbol añade a la suboración la palabra correspondiente al mismo. Este proceso se repite hasta llegar a un *nodo de acción*. Cuando se alcanza uno de estos nodos se añade la palabra correspondiente y se comprueba si es un *nodo de acción válido*. En ese caso se añade la suboración a la oración de respuesta final. En caso contrario se desestima la suboración y se empieza con una nueva suboración vacía.

En todos los casos, cuando se añade una nueva palabra a la suboración o cuando se añade una nueva suboración a la oración de respuesta final, se tiene en cuenta si ya fueron añadidas partes anteriores a ese mismo nivel. En ese caso, antes de añadir la nueva parte (bien sea una palabra o una suboración) se adjunta la conjunción *o* para determinar la disyunción. Además, si anteriormente ya se añadió una conjunción *o* a ese nivel, ésta se cambiar por una coma (,) con el fin de crear una oración más natural.

Supongamos un escenario donde las tres entidades están apagadas y el usuario pronuncia *Enciende, por favor*. El proceso de generación de la oración seguiría los siguientes pasos:

- El MPA inicia el proceso de clarificación y generación desde el nodo *Encender* (el número 3).

- Desde ahí se desciende al nodo *Luz* (el número 6) con una suboración vacía. En ese punto se utiliza el MAP para añadir la palabra *luz* a la suboración. Al ser un nombre común femenino singular y parte objeto se antepone el artículo *la* por lo que se añade a la suboración *la luz*.
- A continuación se desciende al nodo *Fluorescente* (el número 15). Al tratarse de un nombre y una parte modificadora el MAP antepone la preposición *de* y a continuación el artículo determinado masculino singular *el*, contrayéndolo con la preposición. La suboración quedaría de la forma *la luz del fluorescente*. Este es un *nodo de acción válido* por lo que la suboración se añade a la oración de respuesta final, que se encontraba inicialmente vacía.
- En el siguiente paso el MPA pasa al siguiente nodo empezando con una nueva suboración vacía. El siguiente nodo es *Halógena* (el número 16). Se añade a la suboración la palabra *halógena*. Al tratarse de un *nodo de acción válido* se añade la suboración a la oración de respuesta final. Además, como ya existía una suboración añadida se antepone la conjunción *o*. Por lo tanto la oración de respuesta final quedaría de la forma *la luz del fluorescente o halógena*.
- A continuación se desciende al nodo *Radio* (el número 7) y siguiendo un proceso similar se obtendría la suboración *la radio*. Al ser un nodo que genera un subdiálogo se considera un *nodo de acción válido* (ver apartado 5.3.3.1) por lo que esta suboración se añade a la oración de respuesta final. Para ello se sustituye la *o* existente por una coma y se precede la suboración por una nueva conjunción *o*. La oración de respuesta final quedaría de la forma *la luz fluorescente, halógena o la radio*.
- El último nodo (*Lámpara*, el número 8) se desestima al tratarse de un sinónimo de un nodo anterior (el número 6).

Al finalizar se obtiene que el número de acciones válidas es igual a tres, que los *nodos de acción* son *Fluorescente* (el número 15), *Halógena* (el número 16) y *Radio* (el número 7) y que la oración final de clarificación es *la luz del fluorescente, halógena o la radio*. Al haber varias acciones posibles el MPR pronuncia la siguiente oración de clarificación: *Prefieres la luz del fluorescente, halógena o la radio*.

5.3.4.3 Contracción de oraciones ante múltiples acciones posibles

Si el MPR recibe que el número de acciones válidas es superior a tres, éste no pronuncia la oración de clarificación completa, sino que crea una nueva oración más corta basándose en la original. Con esto se consigue que la interfaz no pronuncie una oración con demasiadas opciones, que pueda ser difícil o tediosa de recordar, pero que aún así pueda guiar al usuario en el proceso de interacción.

La nueva oración sólo contendrá la acción solicitada por el usuario y la parte principal de la oración de clarificación. Aún así, la oración de clarificación completa se guarda

para poder ofrecer asistencia en caso de ser necesaria (ver más adelante el apartado 5.3.4.5).

Para reducir la oración se obtiene el primero de los hijos del nodo desde donde se ha iniciado el proceso de clarificación. Dependiendo del tipo de parte a que corresponda se construirá una oración diferente:

- Si el hijo corresponde con una parte de objeto indirecto se inicia la nueva oración con la frase *A quién quieres*, a continuación se adjunta la parte verbal y por último el resto de la oración pronunciada por el usuario.
- Si corresponde con una parte modal la nueva oración empezará con *Cómo quieres*, a continuación seguirá la parte verbal y se continuará con el resto de la oración pronunciada por el usuario.
- En el resto de los casos la nueva oración comenzará con *Qué* seguida de la oración pronunciada por el usuario exceptuando la parte verbal, a continuación la palabra *quieres* y por último la parte verbal de la oración del usuario.

De este modo se consigue una oración que adapta la respuesta al requerimiento realizado por el usuario y que solicita nueva información sin tener que enumerar todas las opciones.

Por ejemplo, para el primero de los casos supongamos que el usuario pronuncia la frase *Quiero llamar por teléfono* y que el número de personas a las que se puede llamar es superior a tres (esto es, debajo del nodo *Teléfono* habrá más de tres nodos con palabras que corresponden con partes de objeto indirecto). En ese caso la oración de clarificación se reduciría de la siguiente manera. Primero se formaría con la frase *A quién quieres* luego se adjunta la parte verbal de la oración del usuario, esto es, *llamar* y a continuación el resto de la oración del usuario, *por teléfono*. Por lo tanto la respuesta que pronunciará la interfaz será *A quién quieres llamar por teléfono*.

La respuesta siempre se adapta a la oración pronunciada por el usuario, empleando las mismas palabras. Si para el caso anterior el usuario hubiera pronunciado *Quiero telefonear* la respuesta generada por el sistema sería *A quién quieres telefonear*. De este modo se obtienen respuestas distintas que se acercan al modo en que el usuario interacciona con el sistema, haciendo que el proceso sea más natural.

Un ejemplo para el tercero de los casos se basa en la Figura 49. Supongamos que debajo del nodo *Luz* (el número 6) hubiera cuatro nodos distintos (en lugar de los dos actuales) que corresponden con cuatro luces diferentes presentes en el entorno. El usuario pronuncia la oración *Quiero que enciendas la luz, por favor*. La interfaz obtiene coincidencias hasta el nodo *Luz* y al realizar el proceso de clarificación obtiene que existen cuatro posibles acciones, por lo que decide acortar la oración de clarificación. Dado que el primer hijo del nodo *Luz* corresponde con una parte objeto inicia la nueva oración con la palabra *Qué*. A continuación adjunta las palabras pronunciadas por el usuario exceptuando la parte verbal. En este caso se adjuntaría la palabra *luz*. A

continuación se adjunta el verbo *quieres* y por último la parte verbal de la oración del usuario, esto es, *encender*. La oración de clarificación que se pronunciará es *Qué luz quieres encender*.

5.3.4.4 Contracción de oraciones con múltiples verbos

Este corresponde con un caso particular de la reducción de oraciones vista en el apartado anterior. Puede que la oración de clarificación generada se refiera a distintas acciones y, por lo tanto, contenga varios verbos. Este caso se suele producir cuando el sistema no conoce la acción que el usuario quiere llevar a cabo, bien porque éste no la ha expresado o porque el reconocedor no la ha obtenido.

La nueva oración se produce concatenando todos los verbos que denotan acciones distintas y a continuación enlazando cada palabra de la oración pronunciada por el usuario.

Por ejemplo, basándonos en la Figura 49 supongamos que el entorno cuenta con una nueva luz (de lectura, por ejemplo) y que el árbol incluye los nodos necesarios para bajar la intensidad de la lámpara halógena (equivalentes a los ya existentes pero precedidos del nodo *Bajar*). La lámpara halógena se encuentra encendida y el resto de las entidades apagadas. El usuario pronuncia (o el reconocedor sólo devuelve) *Luz*. Tras la clarificación se obtiene que el número de posibles acciones es cuatro y una oración de clarificación del tipo *Prefieres encender la luz del fluorescente, encender la luz de lectura, subir la luz halógena o bajar la luz halógena.* Esta oración se reduce empleando una nueva que se forma, en primer lugar, por los verbos de la oración original, esto es, *encender*, *subir* y *bajar*. A continuación se añadirían las palabras pronunciadas por el usuario. En este caso *luz*. La oración de clarificación que se pronunciaría quedaría de la forma *Prefieres encender, subir o bajar la luz.*

Como hemos visto en estos dos apartados, en el caso de que exista un número excesivo de acciones que ofrecer la repuesta se acorta, intentando agilizar el diálogo. Además, la nueva oración se adapta a la forma en que el usuario pronunció la oración. Esto es, si el usuario empleó un verbo el sistema responderá utilizando ese mismo verbo y si se refirió a las entidades de un modo el sistema también responderá refiriéndose a éstas de la misma manera (en lugar de los otros sinónimos posibles).

5.3.4.5 Asistencia al usuario

Cuando se reduce la oración de clarificación, el diálogo se puede volver más ágil para personas familiarizadas con el entorno. Sin embargo esta reducción también puede suponer un problema para usuarios noveles. Si un usuario no sabe cómo dirigirse al entorno ni cómo nombrar a las entidades que se encuentran en el mismo puede llegar a resultar más conveniente utilizar una oración completa, describiendo todas las posibles acciones válidas.

De esta forma la oración de clarificación original se conserva para poder ser utilizada en el caso de que sea necesario (ver apartado 5.3.4.3). Cuando la interfaz responde al usuario con una oración de clarificación simplificada ésta inicia un contador para saber cuánto tarda el usuario en responder. En caso de que se sobrepase un determinado umbral de tiempo el sistema considera que la persona no sabía cómo dirigirse al entorno, por lo que pasa a enumerar la lista completa de acciones posibles.

Con esto se ha comprobado que usuarios noveles que no tenían una idea inicial de cómo dirigirse al entorno han aprendido a interaccionar con el mismo al recibir la información completa en el momento en que lo necesitaban. Por otro lado, usuarios familiarizados con el entorno reciben una respuesta más escueta que les resulta suficiente para continuar el proceso de interacción.

5.3.4.6 Señales de audio

La interfaz no sólo ofrece información al usuario a través de las oraciones generadas en el proceso de clarificación sino que también emplea señales de audio (ver apartado 2.8.1). Estas señales intentan trasmitir un mensaje al usuario de la interfaz de forma rápida y sin necesidad de distraerle en caso de que no sea necesario.

Un ejemplo de utilización de estas señales se produce cuando el reconocedor de voz envía información al supervisor de los diálogos. En ese momento se emite una pequeña señal sonora que permite que los usuarios identifiquen que se ha reconocido una oración. Esta señal informa al usuario del entorno que la interfaz ha recibido una frase, ya que en ocasiones el reconocedor de voz es incapaz de devolver ningún tipo de información tras pronunciar una oración.

Otra situación en la que se han empleado señales de audio se produce cuando el reconocedor de voz retorna al estado de *dormido* (ver apartado 3.9.2.3). Si esta situación se produjo porque el usuario terminó su interacción con el entorno, éste puede haber cambiado su foco de atención, por lo que no es necesario distraerle. En caso contrario, el usuario sigue prestando atención a la interacción con el entorno por lo que una breve señal (similar a un pequeño bostezo) es suficiente para proporcionarle esta información.

Otra señal de audio se ha utilizado cuando el MPN no es capaz de encontrar ninguna coincidencia dentro del árbol lingüístico. Esta señal se repite dos veces seguidas y a la tercera se informa al usuario de que debe cambiar el tipo de oración para que la interfaz pueda reconocer lo que dice. Esta señal intenta evitar molestias al usuario si se producen errores de sustitución en el reconocedor de voz (ver apartados 2.7.2.2 y 2.8.2.3) o si el reconocedor interpreta ruido como una oración del usuario.

Se ha comprobado que la eficacia en el uso de estas señales depende del usuario que esté interaccionando con el entorno. Fundamentalmente, usuarios más avanzados o acostumbrados al sistema de interacción oral perciben con total naturalidad el empleo

de las señales de audio. Sin embargo en usuarios noveles resulta más efectivo informar empleando frases completas.

Por este motivo, durante la fase de evaluación de la interfaz (ver capítulo 6) el sistema sólo utiliza la primera de las tres señales de audio explicadas, aunque se podría considerar la integración de otras señales según aumenta la experiencia del usuario en el entorno o tras una descripción detallada de su funcionalidad.

5.4 Pasos en el diseño de un entorno provisto de una interfaz de diálogos orales

Como se ha visto hasta ahora, son varios los pasos que se deben llevar a cabo en el diseño de los elementos que permiten la creación automática de la interfaz de diálogos orales (ver apartado 5.2). Cada uno de estos pasos se ha encapsulado de modo que pueda ser realizado por personas distintas, que se ocupen de partes concretas y especializadas del diseño. El proceso recaerá fundamentalmente en los diseñadores de nuevas clases de entidades, que implementarán las características necesarias para que entidades de una clase nueva puedan formar parte de los entornos, y en los diseñadores de entornos específicos, que determinarán las características de entornos concretos empleando las clases de entidades ya definidas.

En primer lugar, antes de poder definir una nueva entidad es necesario asegurarse que la entidad sea capaz de comunicarse con la capa *middleware*. Para ello se deben cumplir dos medidas básicas:

- El sistema desarrollado incorpora una API de comunicación de la entidad con la pizarra. Esta API emplea el protocolo HTTP y envía mensajes en formato XML para interaccionar con la pizarra (ver apartado 3.4). Además de la API ya existente se podría proporcionar un conjunto diverso de APIs que permitiera a las entidades comunicarse con la pizarra en diversos lenguajes. El diseñador de un nueva clase de entidad puede emplear la API proporcionada o crear con facilidad una nueva API de comunicación para cualquier otro lenguaje siguiendo el protocolo HTTP y los mensajes XML establecidos.
- Para que una nueva clase de entidad pueda comunicarse con la pizarra sólo tiene que ser capaz de enviar y recibir los mensajes XML mencionados utilizando el protocolo HTTP. De este modo, en el caso de que la aplicación o dispositivo físico que conforma la entidad no permita el uso de estos mensajes, el diseñador de la nueva clase de entidad debe crear una capa intermedia que permita su comunicación mediante ellos (ver apartado 3.1).

A continuación el diseñador de una clase tipo de entidad debe definir una serie de elementos para que entidades de esa clase puedan formar parte de los entornos. Los pasos que se deben seguir son:

- Se debe crear una definición en XML de las propiedades de la clase de entidad, que hará pública para que pueda ser utilizada por todos aquellos que deseen emplear ese tipo de entidades en sus entornos. Esta información se define mediante el DDCE (ver apartado 3.2.1).
- Además de estas propiedades, se puede parametrizar la clase incluyendo nuevas propiedades específicas. Por ejemplo, estas nuevas propiedades se pueden referir la interfaz Web (ver apartado 3.7) o a la interfaz de diálogos orales (ver apartado 5.1.2). Para este caso se emplea el DPCE (ver apartado 3.2.1.1). Esta información adicional se puede añadir por individuos diferentes, cada uno de ellos especializado en un campo de trabajo distinto.
- El diseñador puede necesitar crear los métodos *BBAction* (ver apartado 5.1.3.1) y *hasRequestedADifferentAction* (ver apartado 5.1.3.2). Estos métodos son necesarios para automatizar el proceso de interacción mediante diálogos orales. Como se ha explicado, si la clase de entidad realiza operaciones binarias o si ha heredado sus propiedades de otras clases con las mismas características, no será necesario que el diseñador implemente estos métodos para la nueva clase.
- Para la interfaz de diálogos orales el sistema incorpora una gramática que cubre las necesidades de interacción con las entidades del entorno. En el caso de que esta gramática no se adecue a las características de la clase de entidad que se está definiendo, el diseñador de la nueva clase puede crear un nuevo tipo de gramática, que pueda servir para este y otros tipos de entidad futuros (ver apartado 5.1.4).

Una vez que queda definida una clase de entidad los diseñadores de entornos pueden emplear estas definiciones para configurar los entornos inteligentes que deseen. En este caso, los pasos con que se debe proceder son:

- Se especifica mediante el DDEE (ver apartado 3.2.2) la composición del entorno, a partir de su disposición en espacios y habitaciones y en su división lógica. A continuación se describen en el mismo documento las entidades presentes en el entorno por medio de instancias de las clases de entidad.
- Mediante el DPEE (ver apartado 3.2.2.1) se establecen relaciones entre las entidades los espacios o habitaciones o a algún otro tipo de división lógica. Por último se especifican las posibles relaciones que pudiera haber entre las propias entidades del entorno (ver apartado 3.2.3). Para ello cada entidad empleará la API definida para su clase de entidad. Las entidades pueden incluir información del tipo persona, dispositivo físico, aplicación, etc..
- En aquellas entidades en donde resulte necesario se puede modificar o añadir nueva información en el DPEE para adaptarla a las características concretas del entorno creado (ver apartado 3.2.2.1). Este es el caso de la parametrización de las entidades para la interfaz Web y la interfaz de diálogos orales.

- Con esta información se obtiene un Documento de Descripción del Entorno (DDE) con todas las características del mismo. Este documento permite la creación automática del *middleware* (ver apartado 3.3), la interfaz de control Web (ver apartado 3.7) y la de interacción mediante diálogos orales (ver apartado 5.2).

Cada vez que se cree un nuevo elemento se deberán considerar los puntos establecidos para el diseñador de una nueva clase de entidad. Estos pasos pueden ser realizados por la misma persona o cada uno de ellos puede estar implementado por una persona distinta (por ejemplo personas especializadas en comunicación con dispositivos físicos y en interacción lingüística establecerían parámetros diferentes). En todo caso, la realización de estas tareas se basa en mecanismos estándar que pueden ser llevados a cabo de forma sencilla sin necesidad de tener conocimientos globales sobre todo el sistema. Se busca que sea el proveedor de una nueva clase de entidad el que se ocupe de entregar, junto con la entidad, el conjunto de implementaciones necesario para su incorporación a entornos inteligentes.

Finalmente podrá ser otra persona quien se encargue de la definición de los entornos. Además, esta misma u otras personas (pudiendo dividirse también, dependiendo de su grado de especialización) se podrán encargar, en los casos que sea necesario, de adaptar cada entidad a los entornos concretos. El caso más común es el de la adaptación de las entidades del entorno inteligente a las diferentes interfaces.

5.5 La interfaz de diálogos orales en el entorno inteligente implementado

Dentro del entorno inteligente implementado se ha definido una interfaz que permite interaccionar con las entidades que componen al mismo (ver apartado 3.5).

Esta interfaz de diálogos orales permite interaccionar con las cinco luces del entorno, la cerradura electrónica de la puerta de entrada y un sintonizador de radio con catorce emisoras diferentes. Además se ha simulado la presencia de un aparato de aire acondicionado y de un sistema sencillo para realizar llamadas de teléfono a cuatro posibles personas. Por último existe un módulo que no está asociado a ninguna entidad del entorno que permite establecer saludos, despedidas, agradecimientos, etc. Estos elementos corresponden con siete tipos diferentes de entidades: *luz regulable*, *luz fluorescente*, *luz de foco*, *puerta*, *radio*, *aire acondicionado* y *teléfono* más uno de *buenas maneras*.

Para cada uno de los tipos de entidad se ha definido su información lingüística (ver apartado 5.1.1). Un ejemplo de la definición lingüística que se ha utilizado en la interfaz empleada en el entorno real se presenta en la Figura 50, donde se muestra la definición lingüística de una clase *dimmerlight* y en la Figura 51 donde se representa la parametrización de la instancia *lampv1* de tipo *dimmerlight* empleada en el entorno desarrollado.

```
<class name="dimmerlight">

   <property name="status">
      <paramSet name="dialogue">
         <param name="action1">encender</param>
         <param name="skeleton1"> VP encender poner
         OP luz NCFS000:lámpara MP ambiente halógena </param>
         <param name="action2"> apagar </param>
         <param name="skeleton2"> VP apagar quitar
         OP luz NCFS000:lámpara MP ambiente halógena </param>
      </paramSet>
   </property>

   <property name="value">
      <paramSet name="dialogue">
         <param name="action1">subir</param>
         <param name="skeleton1"> VP subir aumentar
         OP luz NCFS000:lámpara MP ambiente halógena</param>
         <param name="skeleton2"> VP subir aumentar OP intensidad
         LP luz NCFS000:lámpara MP ambiente halógena </param>
         <param name="action2"> bajar </param>
         <param name="skeleton3"> VP bajar disminuir reducir
         OP luz NCFS000:lámpara MP ambiente halógena </param>
         <param name="skeleton4"> VP bajar disminuir reducir
         OP intensidad LP luz NCFS000:lámpara
         MP ambiente halógena </param>
      </paramSet>
   </property>

</class>
```

Figura 50. Información lingüística para una entidad de tipo luz regulable utilizada en el entorno

```
<entity name="lampv1">

   <property name="status">
      <paramSet name="dialogue">
         <param name="add_all">LP izquierda</param>
      </paramSet>
   </property>

   <property name="value">
      <paramSet name="dialogue">
         <param name="add_all">LP izquierda</param>
      </paramSet>
   </property>

</entity>
```

Figura 51. Definición de los parámetros de la instancia lampv1 empleada en el entorno

Una vez definida la información lingüística para cada tipo de entidad y sus métodos *BBAction* y *hasRequestedADifferentAction* asociados (ver apartado 5.1.3) sólo es necesario definir qué entidades se encuentran presentes en el entorno, tal y como se definió en el apartado 3.2.3.

Con estos elementos se crean de forma automática ocho gramáticas distintas, una por cada tipo de entidad (ver apartado 5.2.1). A continuación se construye el árbol lingüístico adaptado al entorno implementado (ver apartado 5.2.2). Este árbol se compone de un total de 310 nodos diferentes. Las combinaciones de caminos existentes en el árbol, la posibilidad de saltar nodos del árbol y el empleo de las gramáticas creadas hacen que las posibles interacciones que la interfaz puede establecer se eleven de forma considerable, permitiendo interacciones múltiples y naturales con las entidades del entorno.

6 Evaluación del sistema propuesto

"No seas siempre riguroso ni siempre blando, y escoge el medio entre estos dos extremos; que en esto está el punto de la discreción"
Don Quijote – Capítulo LI – Segunda Parte

"Siendo poeta podrá ser famoso, si se guía más por el parecer ajeno que por el propio"
Don Quijote – Capítulo XVIII – Segunda Parte

Existen dificultades para evaluar satisfactoriamente el rendimiento de un sistema de diálogos orales dado que la naturaleza de la información que se debe medir resulta compleja y difícil de clasificar. Sin embargo se establecen comúnmente diversos parámetros de evaluación que se pueden clasificar en objetivos y subjetivos. En la primera categoría se tendrían en cuenta el nivel de reconocimiento, nivel de comprensión, número de turnos, número de correcciones necesarias, etc. En la segunda categoría se busca obtener la opinión del usuario sobre distintos aspectos del sistema tales como su utilidad, el nivel de comprensión del mismo, la satisfacción general en la interacción, etc.

Estas pautas son las que se han seguido para realizar la evaluación del sistema de diálogos automático presentado para el entorno inteligente desarrollado.

6.1 Descripción del modelo de evaluación

Para realizar la evaluación se ha construido un modelo evaluación en donde los usuarios debían completar un total de 23 tareas diferentes sobre las entidades del entorno creado (ver apartado 3.5). Estas tareas corresponden con el tipo de interacciones especificadas en el apartado 5.5. Las interacciones se basan en las cinco luces del entorno, la cerradura electrónica de la puerta de entrada y un sintonizador de radio con catorce emisoras diferentes. Además se ha simulado la presencia de un aparato de aire acondicionado y de un sistema sencillo para realizar llamadas de teléfono a cuatro personas diferentes. Por último existe un módulo que no está asociado a ninguna entidad del entorno que permite establecer saludos, despedidas, agradecimientos, etc.

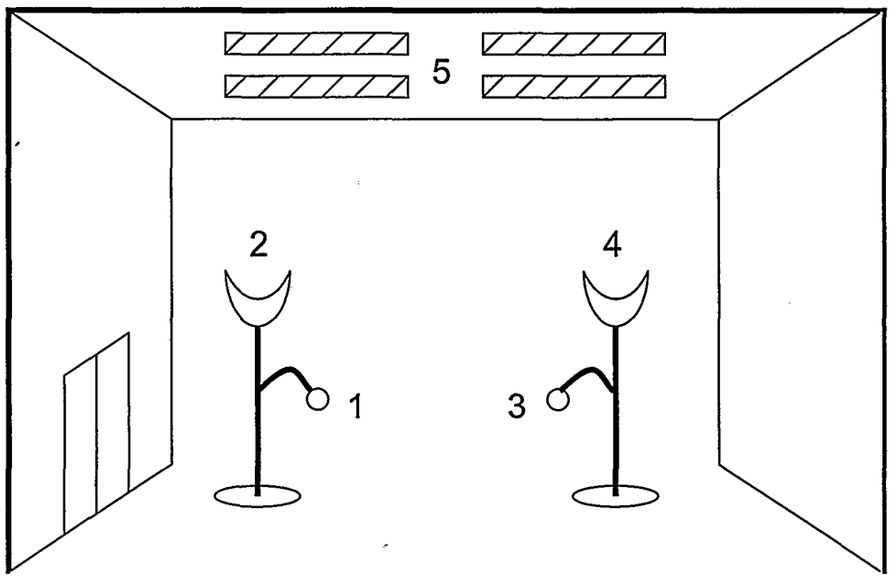


Figura 52. Esquema de las luces del entorno utilizado durante la evaluación

Cada persona realizaba la prueba de forma individual, sin la presencia de ninguna otro individuo, con el fin de evitar que las interacciones de otros usuarios les sirvieran de guía o ayuda. Cada sujeto que realizaba el proceso de evaluación recibía un pequeño mapa del lugar donde aparecía numeradas cada una de las cinco luces que componen el entorno, ver Figura 52. Este pequeño mapa se utiliza para que los usuarios no sepan desde un principio cómo deben interaccionar con cada uno de los elementos del entorno.

Además de este esquema, cada usuario recibía una plantilla con las 23 tareas que debían realizar. En este plantilla se hace referencia a las entidades del entorno de forma genérica, nuevamente con el fin de evitar dar ninguna indicación sobre cómo se debe interaccionar con el entorno. Una trascripción de la plantilla que recibía cada individuo que evaluaba el sistema se muestra en la Figura 53.

```
1 .... Activar 5
2 .... Activar 1
3 .... Desactivar 1
4 .... Activar 4
5 .... + 4
6 .... + 4
7 .... - 4
8 .... Desactivar 4
9 .... Activar 3
10 ... Activar radio
11 ... Activar otra emisora
12 ... Activar 2
13 ... + Radio
14 ... + 2
15 ... Activar puerta
16 ... Desactivar radio
17 ... Activar aire acondicionado
18 ... Aire acondicionado
19 ... Desactivar 2
20 ... Desactivar aire acondicionado
21 ... Desactivar 3
22 ... Llamar Javier
23 ... Desactivar 5
```

Figura 53. Plantilla con las 23 tareas que debía realizar cada usuario

La descripción de cada una de estas tareas y su finalidad en la evaluación es la siguiente:

- Tarea 1. Encender los fluorescentes. Un primer contacto con la interfaz mediante la realización de una tarea sencilla.
- Tarea 2. Encender el foco de la izquierda. Una tarea más complicada que la anterior al poder existir problemas para saber cómo referirse al elemento del entorno. También se pretendía determinar las diferentes formas que la gente emplea para

referirse al mismo elemento, si era necesario recibir asistencia (ver apartado 5.3.4.5) y, en ese caso, si ésta resultaba de utilidad.

- Tarea 3. Apagar la luz recién encendida. Se buscaba comprobar si se utilizaba una anáfora o si se empleaba una oración completa.
- Tarea 4. Encender la luz de ambiente de la derecha. Un nuevo tipo de luz con un nombre diferente.
- Tarea 5. Subir la intensidad de la luz recién encendida. Nuevamente se puede emplear una anáfora, dirigirse a la luz con el nombre completo o sólo con una parte al ser la única entidad del entorno que permite aumentar su estado actual (ver apartado 5.3.3.6).
- Tarea 6. Repetir la acción anterior. Se comprueba si se repite la última oración o se utiliza alguna palabra especial como *Más* (ver apartado 5.3.3.9).
- Tarea 7. El proceso contrario a los anteriores, disminuir la intensidad de la luz.
- Tarea 8. Apagar la luz. Desde la tarea 5 hasta esta tarea 8 ha sido posible utilizar anáforas.
- Tarea 9. Encender el foco de la derecha. Una vez realizada la tarea 2 con una luz similar no debería resultar de dificultad para el usuario.
- Tarea 10. Poner una emisora de la radio. El individuo que evalúa el sistema no recibe instrucciones sobre qué emisoras puede invocar. Se puede utilizar un subdiálogo o realizar la tarea de forma directa.
- Tarea 11. Seleccionar otra emisora. Una vez realizada la tarea anterior es más probable que se evite utilizar el subdiálogo.
- Tarea 12. Encender la luz de ambiente de la izquierda. Esta tarea se ha de realizar con el ruido de fondo de la radio.
- Tarea 13. Subir el volumen de la radio. En caso de no reconocer la oración completa puede ser necesaria la clarificación, ya que hay dos elementos del entorno que pueden aumentar su intensidad (ver apartado 5.3.3.3): la luz de ambiente de la izquierda y el volumen de la radio.
- Tarea 14. Subir la intensidad de la luz de ambiente de la izquierda. La casuística es similar a la tarea anterior pero aplicada sobre la luz.
- Tarea 15. Abrir la puerta de entrada. El timbre de entrada suena durante varios segundos seguidos introduciendo más ruido en el entorno.
- Tarea 16. Apagar la radio. Se elimina la fuente principal de ruido para proseguir con la interacción.

- Tarea 17. Encender el aire acondicionado. Un subdiálogo solicita al usuario la temperatura a la que quiere fijarlo en el caso de que esta no se haya especificado en la interacción.
- Tarea 18. Bajar la temperatura del aire acondicionado. Similar a las acciones de aumentar o disminuir realizadas hasta el momento.
- Tarea 19. Apagar la luz de ambiente de la izquierda. Todavía quedan dos luces encendidas en el entorno.
- Tarea 20. Apagar el aire acondicionado.
- Tarea 21. Apagar el foco de la derecha. No es necesario referirse a ella con su descripción completa.
- Tarea 22. Realizar una llamada de teléfono a Javier.
- Tarea 23. Apagar la luz de los fluorescentes. Al ser la última entidad que permanece activa no es necesario referirse a ella de forma exhaustiva.

Para cada una de las acciones se anotaba las oraciones pronunciadas por el usuario, las oraciones devueltas por el reconocedor (posiblemente en varios turnos), la respuesta sintetizada por el sistema y la acción llevada a cabo. Esta información se anotaba manualmente y mediante un sistema de registro de la interfaz (log) que almacena en ficheros las acciones que acontecen en la misma.

```
1. ¿Consideras útil el uso de diálogos orales para la interacción con
el entorno?

2. ¿Utilizarías este sistema para interaccionar con los entornos en
los que estuvieras?

3. ¿Crees que el sistema ha entendido lo que le has solicitado?

4. ¿Consideras que el diálogo resulta ágil?

5. ¿Has entendido las respuestas proporcionadas por el sistema?

6. ¿Crees que este modo de interacción mediante diálogos orales
resulta más útil que los modos de interacción convencionales?

7. En caso afirmativo, ¿en qué situaciones consideras que puede
resultar de utilidad?

8. ¿Qué añadirías, eliminarías o modificarías para mejorar el sistema
de diálogos?

9. Tu interacción con el sistema ha resultado:
```

Figura 54. Cuestionario que debían rellenar los usuarios del sistema

Una vez finalizada la interacción con el entorno se solicitaba a los usuarios que rellenaran un formulario para conocer el grado de satisfacción y opinión en el empleo de esta interfaz en particular y de este tipo de sistemas en general. El cuestionario al que debían responder se muestra en la Figura 54.

Todas las preguntas son de tipo test exceptuando la 7 y la 8, que permiten una respuesta abierta. Las opciones que se ofrecían al usuario para cada pregunta tipo test son: *Muy satisfactoria, Bastante satisfactoria, Ni satisfactoria ni insatisfactoria, Bastante insatisfactoria y Muy insatisfactoria.*

La finalidad de cada una de las preguntas realizada es la siguiente:

- Pregunta 1. Comprobar el grado de utilidad que presenta a los usuarios un sistema como el utilizado para la interacción con el entorno.
- Pregunta 2. Ver la aceptación de la implantación de una de estas interfaces en los entornos de uso cotidiano.
- Pregunta 3. Centrándose en el sistema concreto que acaban de utilizar, determinar si se considera que la interfaz ha entendido e interpretado correctamente al usuario.
- Pregunta 4. Comprobar la agilidad del sistema utilizado. Este factor depende del nivel de reconocimiento, capacidad de comprensión y número de turnos necesarios.
- Pregunta 5. Determinar el nivel de inteligibilidad de las respuestas automáticas creadas por la interfaz.
- Pregunta 6. Comprobar si se considera que la interfaz presentada constituye una mejora con respecto a otros métodos de interacción convencionales utilizados hasta el momento.
- Pregunta 7. Ver en qué situaciones se considera que resulta de utilidad un sistema como el presentado.
- Pregunta 8. Centrándose en la interfaz utilizada, conocer qué se cambiaría de ella para mejorarla.
- Pregunta 9. Comprobar el nivel de satisfacción general con la interfaz de diálogos orales utilizada para interaccionar con los elementos de un entorno.

6.2 Resultados de la evaluación

La evaluación se ha realizado con 37 usuarios diferentes de forma individualizada. Se ha procurado que la distribución de los individuos que evaluaban el sistema fuera lo más heterogénea posible. Una parte de estas personas posee una formación tecnológica, por lo que está acostumbrada al uso de nuevas fuentes de información, mientras que la otra parte de usuarios no trabajan directamente dentro del ámbito de las tecnologías de la información. Además, las personas que han evaluado el sistema pertenecen a ambos sexos y se encuentran dentro de un amplio rango de edades. Como dato reseñable, más de la mitad de las personas que realizaron la evaluación no tenían relación directa con

el autor del presente estudio siendo éste, en muchos casos, el primer y único contacto que se ha producido entre ambos.

De cualquier manera., las personas que evaluaron el sistema no estaban familiarizadas con el uso de interfaces de diálogos orales avanzadas (más allá de las que se pueden encontrar en un servicio telefónico comercial de acceso a banca o información). El reconocedor de voz no fue entrenado y los usuarios realizaron estas tareas por primera y única vez. Ningún usuario recibió un ejemplo verbal previo que les permitiera intuir cómo podían interaccionar con el entorno.

Cada uno de los usuarios tenía que completar de forma secuencial las 23 tareas especificadas en la Figura 53. Ellos mismos podían comprobar los resultados obtenidos ya que todas sus interacciones tenían consecuencias físicas dentro del entorno (o respuestas orales en el caso de los actos simulados). Posteriormente rellenaban el formulario representado en la Figura 54.

Con los datos obtenidos se obtienen los resultados de los parámetros objetivos y subjetivos de evaluación.

6.2.1 Parámetros objetivos de evaluación

Los 37 usuarios del sistema realizaron, en su mayoría, las 23 tareas propuestas. Sólo en algunos casos aislados la persona dejó de realizar alguna tarea, fundamentalmente debido a que se saltaba por despiste alguno de los pasos propuestos en el guión mostrado en la Figura 53. Cuando esto ocurría, se continuaba con el proceso y el supervisor de la evaluación realizaba la tarea que se había dejado pendiente utilizando la interfaz Web (ver apartado 3.7). Solamente hubo dos situaciones donde los individuos se sintieron incapaces de realizar una de las tareas especificadas y se les pidió que continuaran con el resto de la evaluación.

Los datos de evaluación se tomaron a través de los ficheros de registro (log) y de la anotación manual de las interacciones. En total, se tuvieron en cuenta los dieciséis parámetros siguientes:

- Parámetro #1. El número de tareas realizadas por cada individuo.
- Parámetro #2. El número de oraciones pronunciadas por el usuario para completar todas las tares.
- Parámetro #3. Número de oraciones pronunciadas que incluían una anáfora (ver apartado 5.3.3.8).
- Parámetro #4. Número de oraciones basadas en palabras especiales del tipo más/menos (ver apartado 5.3.3.9).
- Parámetro #5. Número de oraciones pronunciadas donde el reconocedor de voz fue incapaz de retornar ningún valor. Esto puede ocurrir porque el usuario pronuncia una frase que se encuentra fuera de los límites de las gramáticas de interacción con el sistema (ver apartado 5.2.1) o (mucho más frecuentemente) porque aunque el

usuario haya pronunciado una frase admitida por el reconocedor, éste ha sido incapaz de obtener ningún resultado.

- Parámetro #6. Tras un oración no reconocida (caso #5), número de oraciones que el usuario vuelve a repetir la misma frase textual al sistema.
- Parámetro #7. Tras las oraciones no reconocidas (caso #5), número de oraciones que el usuario modifica con respecto a la que el sistema no reconoció (la suma de #6 y #7 debe ser igual a #5).
- Parámetro #8. Número de oraciones donde el reconocedor sólo ha obtenido parte de la oración pronunciada por el usuario.
- Parámetro #9. Para las oraciones donde el reconocedor sólo ha devuelto parte de la frase pronunciada (caso #8), en cuántas el sistema ha podido generar una oración de clarificación correcta que haya guiado al usuario en el diálogo (ver apartados 5.3.3.5 y 5.3.3.6).
- Parámetro #10. Para las oraciones donde el reconocedor sólo ha obtenido parte de la frase pronunciada (caso #8), en cuántas oraciones el sistema ha sido incapaz de realizar ningún tipo de interpretación o clarificación (ver apartado 5.3.3.2).
- Parámetro #11. Para las oraciones donde el reconocedor sólo ha entendido parte de la frase (caso #8), en cuántas oraciones el sistema ha sido capaz de interpretar completamente la oración pronunciada y llevar a cabo la acción correspondiente (ver apartado 5.3.3.3). La suma de #9, #10 y #11 debe ser igual a #8.
- Parámetro #12. El número de oraciones donde el sistema ha reconocido toda la frase pronunciada por el usuario pero éste no ha proporcionado toda la información necesaria para realizar una tarea.
- Parámetro #13. Para las oraciones donde el usuario sólo ha proporcionado parte de la información (caso #12), en cuántas oraciones el sistema ha proporcionado una respuesta de clarificación que ha permitido continuar con el diálogo.
- Parámetro #14. Para las oraciones donde el usuario sólo ha proporcionado parte de la información (caso #12), en cuántas oraciones el sistema ha sido capaz de interpretar completamente la oración pronunciada y llevar a cabo la acción correspondiente. La suma de #13 y #14 debe ser igual a #12.
- Parámetro #15. Oraciones donde el reconocedor ha devuelto alguna palabra que el usuario no ha pronunciado o donde ha considerado el ruïdo como una frase del usuario (ver apartado 2.7.2.2).
- Parámetro #16. Casos en donde este reconocimiento erróneo ha provocado que se haya realizado una pregunta de clarificación o una acción incorrecta.

Los valores de cada uno de estos 16 parámetros para los 37 individuos que evaluaron el sistema se muestran en la Tabla 1.

	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16
#1	2	3	5	1	3	3	0	4	2	1	1	4	2	2	1	1
#2	23	38	3	0	5	3	2	7	3	0	4	1	1	0	1	0
#3	21	37	0	3	3	3	0	6	2	2	2	4	2	2	0	0
#4	16	50	0	1	18	12	6	5	2	1	2	3	2	1	0	0
#5	23	43	0	0	12	12	0	4	3	0	1	3	3	0	1	0
#6	23	47	2	3	11	3	8	8	4	3	1	5	3	2	0	0
#7	20	35	0	0	4	0	4	6	5	0	1	3	3	0	1	0
#8	23	36	0	0	7	7	0	8	3	2	3	1	1	0	2	0
#9	22	64	1	0	26	24	2	4	2	1	1	6	4	1	6	5
#10	23	44	2	3	12	8	4	4	0	1	3	2	1	1	1	1
#11	23	48	4	2	14	7	7	4	1	0	3	8	3	5	1	0
#12	23	52	0	0	8	3	5	10	6	2	2	8	4	4	1	1
#13	22	45	0	1	14	8	6	2	1	0	1	2	2	0	1	0
#14	23	52	1	3	15	3	12	3	3	0	0	6	4	2	1	0
#15	23	53	1	3	19	7	12	9	6	2	1	4	2	2	2	1
#16	23	69	0	3	33	10	23	8	4	3	1	4	2	2	1	0
#17	23	50	0	0	15	9	6	3	1	1	1	11	5	7	1	0
#18	23	33	0	0	3	2	1	5	2	0	3	6	3	3	0	0
#19	22	54	0	0	13	7	6	7	6	1	0	12	7	5	0	0
#20	23	56	0	0	20	13	7	9	4	1	4	2	1	1	0	0
#21	23	58	3	2	26	17	9	2	1	0	1	8	5	3	0	0
#22	22	39	0	0	7	2	5	4	2	0	2	7	4	3	0	0
#23	23	62	0	1	14	2	12	10	8	0	2	9	5	4	2	2
#24	23	40	1	0	8	4	4	4	4	0	0	7	2	5	1	1
#25	23	52	0	0	21	20	1	9	5	2	2	3	2	1	0	0
#26	23	51	2	0	15	8	7	8	7	0	1	5	2	3	1	1
#27	22	38	1	0	7	5	2	2	2	0	0	7	3	4	0	0
#28	23	40	0	0	9	2	7	5	3	0	2	3	3	0	0	0
#29	23	46	0	0	10	7	3	4	3	0	1	10	5	5	0	0
#30	23	37	0	0	3	2	1	10	4	2	4	4	2	2	0	0
#31	23	55	0	3	15	11	4	3	1	2	0	7	6	1	1	1
#32	23	63	3	1	19	17	2	4	2	0	2	14	9	5	2	2
#33	23	53	0	0	18	10	8	14	5	3	6	4	2	2	0	0
#34	23	42	0	0	11	5	6	6	4	0	2	0	0	0	0	0
#35	23	36	2	1	3	3	0	2	1	0	1	9	6	3	1	0
#36	23	56	0	3	23	20	3	9	3	3	3	3	1	2	0	0
#37	23	32	5	1	3	2	1	4	2	0	2	3	3	0	2	0

Tabla 1. Parámetros de evaluación objetiva para cada uno de los usuarios

Sobre estos datos se pueden obtener algunos resultados que permiten determinar la funcionalidad y rendimiento del sistema.

En primer lugar, los usuarios han realizado un total de 834 tareas, para lo que han debido formular 1743 oraciones. Por lo tanto, el número medio de oraciones necesarias para realizar una tarea es de aproximadamente 2. Ver Tabla 2.

Tareas realizadas	Oraciones pronunciadas	Oraciones pronunciadas por tarea
834	1743	2

Tabla 2. Tareas realizadas y oraciones pronunciadas por los usuarios

De entre las oraciones pronunciadas por los usuarios, el reconocedor ha sido incapaz de devolver algún valor para 467 de ellas, un 27% de total, ha devuelto sólo parte de la oración pronunciada en 216 oraciones, un 12% del total y ha devuelto información errónea en 31 oraciones, un 2%. Esto implica que la tasa total de errores del reconocedor en las oraciones devueltas se eleva al 14% del total. Ver Tabla 3.

Oraciones donde el reconocedor no ha devuelto ningún valor	Oraciones donde el reconocedor ha devuelto menos información	Oraciones donde el reconocedor ha devuelto algún valor erróneo	Total de oraciones donde el reconocedor ha devuelto información incompleta o errónea
467	216	31	247
27 %	12 %	2 %	14 %

Tabla 3. Oraciones donde el reconocedor de voz ha actuado erróneamente

Como se ha especificado anteriormente, una parte de estos errores se debe a que los usuarios han pronunciado oraciones que están fuera del vocabulario generado para el entorno. Sin embargo, la mayor parte de los errores se producen con oraciones que el sistema admite pero el reconocedor de voz no es capaz de procesar correctamente. Se debe tener en cuenta que se utiliza un reconocedor de voz general que no ha sido entrenado para adaptarse a los usuarios y el entorno, ver apartado 3.9.2.

De entre las oraciones donde el reconocedor sólo ha podido devolver parte de la información pronunciada, en 117 ocasiones la interfaz ha sido capaz de realizar una interpretación parcial de las oraciones y ha formulado una pregunta de clarificación adecuada, un 54% del total, y en 66 oraciones ha sido capaz de interpretar completamente la solicitud y realizar directamente una acción, un 31%. Esto significa

que en un 85% de las ocasiones la interfaz ha podido realizar una interpretación parcial o total de la oración pronunciada por el usuario. En otras 33 ocasiones la interfaz ha sido incapaz de realizar ninguna interpretación sobre la oración pronunciada, un 15% del total. Ver Tabla 4.

Oraciones donde la interfaz ha interpretado parte de la información	Oraciones donde la interfaz ha interpretado completamente la oración	Total de oraciones que se han interpretado parcial o totalmente	Oraciones donde la interfaz ha sido incapaz de interpretar la oración
117	66	183	33
54 %	31 %	85 %	15 %

Tabla 4. Comportamiento de la interfaz ante oraciones donde el reconocedor de voz sólo ha devuelto parte de la información

Entre las oraciones donde el reconocedor ha devuelto alguna parte equivocada (o donde se ha interpretado el ruido como una oración) en 16 ocasiones esto ha generado que el sistema realizara una pregunta de clarificación o una acción errónea, un 51 % del total. En otras 15 oraciones la interfaz ha podido recuperarse del error y continuar con el proceso de forma adecuada, un 49 %. Ver Tabla 5.

Oraciones que han provocado una interpretación errónea	Oraciones donde la interpretación de la interfaz no se ha visto afectada
16	15
51 %	49 %

Tabla 5. Comportamiento de la interfaz ante oraciones donde el reconocedor de voz ha devuelto información errónea (no pronunciada por el usuario)

Para las oraciones donde el reconocedor ha devuelto toda la oración pronunciada por el usuario correctamente (1029 frases en total), en 198 ocasiones el usuario sólo ha proporcionado parte de la información necesaria, un 19% del total. Ver Tabla 6. Esto implica que la mayor parte de las ocasiones los usuarios proporcionan toda la información relativa a la acción que se quiere llevar a cabo en la misma oración. Un aspecto fundamental para que se produzca este hecho se halla en el empleo de la asistencia del sistema (ver apartado 5.3.4.5) además del uso de respuestas de clarificación que guían al usuario sobre cómo debe interaccionar con el entorno (ver apartado 5.3.4.2), sirviéndole de referencia para futuras oraciones.

Oraciones donde el usuario sólo proporciona parte de la información	Oraciones donde el usuario proporciona toda la información necesaria en la misma frase
198	831
19 %	81 %

Tabla 6. Información proporcionada por los usuarios en oraciones reconocidas completamente

Para las oraciones donde el usuario sólo proporciona parte de la información, en 115 ocasiones el sistema ha generado una pregunta de clarificación que ha continuado con el diálogo, un 58% del total. En otras 83 oraciones el sistema ha sido capaz de interpretar complemente la oración a pesar de la carencia de información dada por el usuario, un 42% del total. Ver Tabla 7. Esto se debe fundamentalmente al uso de la información de contexto físico del entorno almacenada en la pizarra (ver apartado 3.3), que permite interpretar las solicitudes del usuario dada su situación actual (ver apartados 5.3.3.5 y 5.3.3.6).

Oraciones que han producido una pregunta de clarificación	Oraciones que se han podido interpretar completamente
115	83
58 %	42 %

Tabla 7. Comportamiento de la interfaz ante oraciones correctamente reconocidas donde el usuario sólo proporciona parte de la información necesaria

Si se unen los valores de las tablas anteriores se obtiene que de entre las 1276 oraciones recibidas por la interfaz en 414 ocasiones estas oraciones sólo contenían parte de la información necesaria para realizar una acción. De entre estas oraciones, en 232 ocasiones la interfaz ha realizado una pregunta de clarificación para continuar con el diálogo, en otras 149 ocasiones la interfaz ha interpretado completamente la oración y ha realizado la acción apropiada y en 33 ocasiones la interfaz no ha podido interpretar la oración. Esto significa que en el 92% de las oraciones recibidas donde faltaba parte de la información la interfaz ha sido capaz de realizar una interpretación parcial o total de la información. Por el contrario, el 8% de estas oraciones no han conseguido ser interpretadas adecuadamente. Ver Tabla 8.

Oraciones que han generado una respuesta de clarificación	Oraciones que se han interpretado completamente	Número total de oraciones interpretadas parcial o totalmente	Oraciones donde no se ha podido realizar ninguna interpretación
232	149	381	33
56 %	36 %	92 %	8 %

Tabla 8. Comportamiento de la interfaz para las oraciones donde se ha recibido parte de la información necesaria para realizar una acción

Por último se ha medido el empleo de anáforas (ver apartado 5.3.3.8) y de frases especiales del tipo más/menos (ver apartado 5.3.3.9). Cada usuario podía utilizar la anáfora en 6 acciones distintas. En total los usuarios han empleado 36 anáforas lo que implica que se han utilizado en un 16% de las ocasiones posibles. Las palabras más/menos se podían haber sido empleada por cada usuario en otras 6 ocasiones. Su uso se ha producido en total en 35 oraciones, por lo que se han utilizado en otro 16% de las situaciones posibles. Ver Tabla 9.

Empleo de anáforas	Empleo de palabras especiales
36	35
16 %	16 %

Tabla 9. Uso de anáforas y palabras especiales

De los resultados mostrados se detectan dos problemas principales (ver apartado 2.7.2.2):

- La dificultad del sistema para recuperarse de los errores de inserción.
- El alto porcentaje de oraciones donde el reconocedor produce errores de reconocimiento (errores de sustitución) o es incapaz de reconocer la oración pronunciada (errores de rechazo).

En el primero de los errores la mitad de las oraciones producen una interpretación errónea por parte del sistema. Sin embargo, este efecto se ve minimizado por el hecho de que las oraciones devueltas con este tipo de error corresponden con aproximadamente el 2% de las oraciones totales.

En el segundo de los casos los porcentajes de recuperación del sistema ante estos errores son altamente satisfactorios. Para este tipo de problemas se consigue interpretar correctamente de forma parcial o total más de ocho de cada diez oraciones con errores.

Por último, cabe destacar que gracias al uso del contexto físico el sistema es capaz de interpretar completamente más de cuatro de cada diez oraciones donde el usuario sólo ha proporcionado parte de la información necesaria a priori.

6.2.2 Parámetros subjetivos de evaluación

Los parámetros subjetivos de evaluación se obtienen a partir de las respuestas del formulario que rellenaba cada uno de los 37 usuarios del sistema tras su interacción con el entorno, ver Figura 54.

Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	Pregunta 6	Pregunta 9
5	4	4	4	4	5	4
4	4	4	3	5	4	4
5	5	4	3	4	4	4
5	5	3	4	5	5	5
5	3	4	4	5	4	4
5	5	2	2	4	2	4
5	5	4	5	5	5	5
5	5	5	3	5	3	5
5	3	4	4	3	5	4
4	5	4	4	5	4	5
5	4	4	4	5	4	5
5	4	4	4	4	5	5
4	4	4	3	5	5	4
4	4	4	3	5	4	5
5	5	4	4	5	5	5
4	4	3	3	3	4	3
4	4	4	4	4	3	4
3	4	4	2	4	2	4
5	5	5	4	5	4	5
5	4	4	4	5	3	4
5	5	4	3	5	5	5
5	5	4	4	5	4	4
5	4	4	3	4	4	4
5	5	4	4	4	4	5
4	4	4	4	5	3	4
4	5	4	4	4	5	4
3	4	4	3	4	2	4
5	5	4	4	4	4	5
5	4	3	4	5	5	5
4	4	5	5	5	4	5
4	5	4	4	5	4	4
5	4	3	4	4	5	5
5	4	4	4	5	3	4
5	4	5	4	5	4	5
4	5	4	5	5	5	5
5	4	5	5	5	5	5

Tabla 10. Respuestas proporcionadas por los usuarios al cuestionario

En todos los casos, con el fin de obtener el mayor grado de sinceridad en la evaluación, las respuestas proporcionadas en el cuestionario eran de carácter anónimo y se insistía de forma especial a los usuarios de la interfaz en que utilizaran un espíritu crítico en sus contestaciones, que permitiera detectar los fallos del sistema y desarrollar las posibles mejoras.

El formulario se componía de nueve preguntas, siete de ellas con respuesta tipo test y otras dos abiertas. A cada respuesta se le ha asignado posteriormente una puntuación, lo que permite obtener estadísticas de los resultados. Las posibles respuestas y los valores asignados son *Muy satisfactoria* (5), *Bastante satisfactoria* (4), *Ni satisfactoria ni insatisfactoria* (3), *Bastante insatisfactoria* (2) y *Muy insatisfactoria* (1). Todos los usuarios han respondido a las siete preguntas tipo test. Las dos preguntas abiertas eran de carácter opcional.

Las respuestas dadas por cada usuario a las siete preguntas tipo test se muestran en la Tabla 10.

A partir de estos valores se puede obtener una media de las apreciaciones hechas por los usuarios (valoradas del 1 al 5) y un porcentaje de aceptación de la afirmación realizada en cada pregunta (tomando valores de 0 al 4). Estos datos se muestran en la Tabla 11.

	Pr. 1	**Pr. 2**	**Pr. 3**	**Pr. 4**	**Pr. 5**	**Pr. 6**	**Pr. 9**
Media	4,58	4,36	3,97	3,75	4,55	4,05	4,47
Porcentaje	88 %	82 %	73 %	68 %	87 %	75 %	85 %

Tabla 11. Medias y porcentajes de aceptación de cada pregunta formulada

Se puede comprobar que en general los usuarios se encuentran altamente satisfechos de las características de la interfaz. Las preguntas sobre la utilidad de los diálogos orales, las posibilidades de uso del sistema y la comprensión de las respuestas obtienen rangos de aceptación superiores al 80%. Las preguntas sobre las capacidades de comprensión del sistema y de las ventajas de la interacción con respecto a medios convencionales se encuentran por encima del 70% de aceptación. La puntuación más baja se obtiene en la agilidad del diálogo, fundamentalmente debido a los fallos del reconocedor. Aún así, la agilidad de los diálogos tiene un rango de aceptación del 68%. Por último, al satisfacción en la interacción con el sistema es de un 85%, un valor elevado.

La primera de las dos preguntas abiertas (¿en qué situaciones consideras que puede resultar de utilidad?) es la que ha recibido un mayor número de respuestas, un total de 36. Éstas se pueden agrupar básicamente en cuatro bloques:

- El sistema puede resultar útil para procesos de interacción remota, donde el usuario se encuentra alejado del entorno de interacción. Han sido 5 los usuarios que han proporcionado esta respuesta, constituyendo un 14% del total de respuestas.
- Puede ser especialmente interesante para ancianos, personas con discapacidades y, en general, para cualquier usuario con dificultades temporales o permanentes de movilidad. Son 19 usuarios los que respondieron en este sentido, correspondiendo con un 53% de las respuestas.
- Presenta ventajas para facilitar la interacción con el entorno, mejorando la calidad de vida dentro del mismo. Ha habido 10 respuestas en este sentido, lo que equivale a un 28%.
- Es especialmente útil en un entorno laboral, donde el mismo individuo es el principal ocupante del espacio que le rodea y debe interactuar con numerosos elementos. Esta respuesta fue proporcionada por 2 personas, un 5% del total.

Interacción remota	**Ancianos y discapacitados**	**Comodidad**	**Entorno laboral**
5	19	10	2
14 %	53 %	28 %	5 %

Tabla 12. Situaciones en las que se esta interfaz de interacción oral se considera de utilidad

Como se puede apreciar en la Tabla 12 la mayoría de las situaciones en donde los usuarios consideran que el sistema puede resultar útil tienen asociados conceptos de movilidad, bien sea por problemas de discapacidad o simplemente por comodidad. En otros casos también se ha apuntado su función para permitir una interacción remota con el entorno o dentro del espacio de trabajo.

Para la segunda de las preguntas abiertas (¿qué añadirías, eliminarías o modificarías para mejorar el sistema de diálogos?) se han obtenido 20 respuestas, con una mayor diversidad de criterios. Las respuestas pasan por dotar de más vocabulario al sistema (#1), proporcionarle de mayor personalidad haciendo que se adapte a cada usuario (#2), mejorar el reconocimiento de voz (#3), ofrecer más información sobre el estado del diálogo (#4), emplear micrófonos de ambiente que no se hayan de portar (#5), permitir que el sistema sea proactivo (#6) y dotarle de reconocimiento de gestos (#7). Ver Tabla 13.

#1	#2	#3	#4	#5	#6	#7
3	3	4	4	3	2	1
15 %	15 %	20 %	20 %	15 %	10 %	5 %

Tabla 13. Aspectos que se modificarían o añadirían al sistema evaluado

6.3 Escalabilidad del sistema

Uno de los mayores problemas en el uso de árboles es que su tamaño crece exponencialmente según aumenta el grado (el número de hijos) de los nodos y la profundidad del árbol.

Dado cómo se construye el árbol, el grado de sus nodos dependerá del número de palabras distintas que se utilicen en el sistema. A mayor número de palabras, mayor será el tamaño del mismo. El número de estas palabras depende de dos factores: la cantidad de entidades distintas presentes en el entorno y la cantidad de sinónimos que se emplee en cada parte lingüística de estas entidades.

El caso peor sería el de un árbol donde todos sus nodos tienen grado k. El número de nodos de un árbol de estas características con profundidad p se muestra en la Figura 55.

$$\sum_{i=1}^{p} k^{i-1}$$

Figura 55. Tamaño de un árbol con nodos de grado k y profundidad p

Este sería el caso, por ejemplo, de una entidad con una acción formada por cinco partes y en la que cada una contiene cuatro sinónimos. Esta información lingüística daría lugar a un árbol de grado 4 y nivel 6, por lo que se crearía un número total de 1365 nodos.

En el equipo donde se localiza la interfaz de diálogos (de tipo medio) la interfaz se crea y funciona correctamente con árboles de hasta aproximadamente 150.000 nodos. Esto significa que se permitirían unas 110 acciones como las descritas en el ejemplo anterior. Suponiendo que cada entidad contiene una media de tres acciones, sería posible crear la interfaz y controlar unas 33 entidades distintas en el entorno.

Sin embargo estos datos están basados en un caso peor de muy difícil ocurrencia en un entorno real. Aunque el entorno se componga de entidades diferentes (algunas muy dispares con respecto a otras) es muy probable que muchas de estas entidades compartan las mismas partes lingüísticas. Si diferentes entidades comparten partes lingüísticas al mismo nivel, no se crearán nuevos caminos en el árbol, sino que se emplearán los nodos ya creados. Esto supone que no se creen árboles completos, reduciéndose considerablemente el número de nodos.

Además, es muy probable que estas coincidencias se den en las partes lingüísticas iniciales, evitando que se aparezcan nuevos nodos en los niveles superiores del árbol (que son los que pueden aumentar considerablemente el tamaño del árbol al desplegar numerosos hijos).

Basándonos en estas ideas podemos suponer el caso de un entorno con múltiples entidades en donde todas inician su información lingüística con una parte verbal de

acción (aunque ya se ha visto que esto no es necesario). En total las entidades permiten utilizar n palabras diferentes para representar sus acciones. A partir de ahí, se puede volver a tomar el caso peor donde todos los nodos restantes tienen grado k. Si el número de partes lingüísticas de cada entidad viene dado por p-1 (por lo que el nivel del árbol es p), el número total de nodos del árbol se especifica en la Figura 56.

$$1+\sum_{i=2}^{p} nk^{i-2}$$

Figura 56. Tamaño de un árbol de nivel p, n nodos iniciales y grado k para los demás nodos

Este sería el caso de un entorno con 100 entidades que en total permitieran realizar 50 acciones diferentes. Si estas 50 acciones se componen de 400 palabras distintas (4 sinónimos por acción), el resto de las partes de cada entidad está formada por 3 sinónimos cada una y las entidades poseen una media de 6 partes se crearía un árbol con 145.601 nodos, manejable por el sistema.

Estos datos permiten comprobar que la interfaz es capaz de crearse y trabajar con entornos de tamaño medio (30 entidades en el caso peor) a medio-grande (100 entidades). Sin embargo no es posible su utilización en entornos de mayores dimensiones o con numerosas entidades. Para solucionar esto existen dos posibles alternativas:

- Cada vez que aparece un nuevo sinónimo (que no estuviera presente en el árbol a ese nivel) se desdoblan los caminos subsiguientes del árbol, creando una nueva ruta a partir del nuevo nodo sinónimo. Esta división de caminos se realiza porque no todos los nodos sinónimos necesariamente contienen las mismas entidades relacionadas y, a pesar de ser inicialmente sinónimos, pueden poseer hijos diferentes. Este sería el caso, por ejemplo, de una entidad donde tendría sentido que *encender* y *dar* fueran sinónimos, mientras que en otra los sinónimos serían *encender* y *poner* (en este caso *dar* y *poner* no serían sinónimos entre sí). Sin embargo, este fenómeno no se produce en todas las ocasiones, por lo que se podrían desdoblar los nodos sinónimos sólo en aquellas situaciones donde fuera realmente necesario. Con esta característica adicional se podría ahorrar la creación de numerosos nodos innecesarios que presentan información repetida.
- Incluso en aquellos casos en los que resulta necesario desdoblar los caminos del árbol, se pueden producir repeticiones de información en los nodos inferiores. Para evitar esto, en lugar de utilizar una estructura exclusivamente en árbol se podría emplear una estructura de grafo (sin ciclos) que permita compartir la información en aquellos niveles en donde sea posible. Esta estructura en ciclo debería prestar especial atención a que los nodos convergentes se refirieran a las mismas entidades y a desdoblar sus caminos en el momento en que uno de estos nodos presente información adicional exclusiva.

7 Conclusiones

"Es mejor ser loado de los pocos sabios que burlado de los muchos necios"
Canónigo – Capítulo XLVIII – Primera Parte

7.1 Aportaciones

La investigación realizada por el autor durante estos últimos años en entornos inteligentes y sistemas de diálogos orales, ha generado una serie de resultados y aportaciones que se han explicado y desarrollado en el presente trabajo.

Una primera aportación, tal y como se puede comprobar en el capítulo 2 y en el amplio trabajo [Montoro, G. 2000], ha sido la realización de un detallado estudio de los entornos inteligentes existentes, mostrando analogías, diferencias, enfoques y aproximaciones y analizando las distintas modalidades de interacción con los mismos. Este estudio no resulta trivial, ya que dada la juventud de esta área de investigación son pocas las formalizaciones realizadas sobre el tema.

Como consecuencia de este estudio se ha diseñado un modelo de entorno inteligente en el que se definen las características requeridas para el desarrollo de uno de estos espacios. Entre estas características se ha concluido que resultan necesarias: una red de control de dispositivos, una red de información multimedia, un *middleware* y las aplicaciones de alto nivel e interfaces.

Una aportación significativa ha consistido en el desarrollo de un entorno inteligente real que ha servido como banco de pruebas para la investigación y desarrollo de nuevas aproximaciones, ideas, aplicaciones e interfaces en entornos inteligentes. Este entorno no sólo tiene su interés para el grupo de investigadores que lo ha desarrollado, sino que puede ser fácilmente empleado por otros investigadores interesados en trabajar con entornos inteligentes.

Este entorno inteligente se ha desarrollado con la implementación de los siguientes elementos:

- Un bus domótico para el control de los dispositivos del entorno (ver apartado 3.1). El bus elegido ha sido EIB (European Installation Bus) que pugna por establecerse como el estándar europeo de buses de control domótico. Mediante este bus se controlan las luces, dispositivo de apertura de puerta, control de tarjetas inteligentes, etc.
- Un bus específico para el control del flujo multimedia (ver apartado 3.1). La información multimedia se envía mediante una red Ethernet, elegida por su extensión y disponibilidad. Por esta red se envía fundamentalmente señal de audio.
- Un *middleware* que aísla a las aplicaciones de los detalles físicos de los dispositivos del entorno (ver apartado 3.3). Las aplicaciones que quieran acceder a los elementos del entorno sólo tendrán que comunicarse con este *middleware*.
- Una serie de aplicaciones inteligentes, entre las que se encuentra una aplicación de acceso al entorno, otra de recibimiento y muestra de información personalizada, una interfaz de control Web, etc.
- Una interfaz de diálogos orales que permite controlar e interactuar con los elementos del entorno de forma dinámica.

El *middleware* constituye una pieza clave en la investigación en el campo de los entornos inteligentes. Se ha desarrollado como una capa intermedia entre las interfaces y aplicaciones y el entorno físico, que contiene una representación de los elementos del entorno y su distribución dentro del mismo. Asimismo, mantiene actualizada en todo momento la información sobre el estado de los elementos en el entorno. Su creación se realiza de forma automática mediante un Documento de Definición del Entorno (DDE) donde se describen la distribución del mismo, los elementos que lo componen y sus características.

Las ventajas en el empleo de esta capa *middleware* son:

- Permite sustituir unos dispositivos por otros de igual funcionalidad sin que sea necesario modificar las aplicaciones que interactúan con el entorno.
- El *middleware* es un repositorio que posibilita conocer cómo está formado el entorno y el estado actual de todos sus elementos. A su vez, proporciona mecanismos estándar para modificar el estado del entorno, aislando a las aplicaciones del manejo de información de bajo nivel.
- Permite desarrollar nuevas aplicaciones de forma fácil y rápida al utilizar estándares como el protocolo http y mensajes XML.

Esta capa *middleware* ha sido complementada con la información necesaria para la creación automática de una interfaz de diálogos orales. Esta constituye otra de las principales aportaciones del presente trabajo. Las ventajas que presenta esta integración son:

- Las instrucciones necesarias para la creación de la interfaz se especifican en el mismo documento y con la misma sintaxis estándar que se utiliza para describir las entidades del entorno. Sin embargo, ambos procesos son independientes y pueden llevarse a cabo en momentos distintos por diseñadores diferentes.
- La información lingüística se asocia a tipos genéricos de entidades por lo que resulta suficiente, en muchas ocasiones, con definir qué elementos componen el entorno para obtener la interfaz de diálogos orales. Asimismo esta información se puede parametrizar para adaptarse a las características de un entorno concreto.
- La interfaz de diálogos orales se crea al mismo tiempo que el *middleware* por lo que se obtiene de forma automática una interfaz adaptada a cada entorno concreto.
- La información que se adjunta a esta capa no se ciñe exclusivamente una interfaz de diálogos orales sino que permite crear otros tipos de interfaces permitiendo dotar al entorno de multimodalidad. Tal es el caso de una interfaz Web desarrollada, que se crea y ejecuta de forma concurrente junto con la interfaz oral.

A partir de estas premisas se han desarrollado unas interfaces de diálogos orales para entornos inteligentes que aportan datos y soluciones en el campo de investigación tratado en el presente trabajo.

Inicialmente se desarrolló un primer sistema de diálogos orales que consistía en un modelo basado en tareas donde se establecían los diálogos según la tarea que debían desarrollar (ver capítulo 4). Este sistema fue probado con público no entrenado y sirvió para conocer las necesidades de una interfaz de diálogos orales en un entorno de estas características. La experiencia obtenida en su desarrollo y las pruebas realizadas hicieron descartar esta interfaz, y nos han servido para demostrar que un requisito fundamental en una interfaz oral para un entorno inteligente es su capacidad para poder configurarse y adaptarse automáticamente al entorno en que se encuentra.

A partir de estas ideas y de la experiencia obtenida se ha realizado el estudio y desarrollo de una interfaz de diálogos orales que se crea de forma automática dependiendo de la configuración actual del entorno en el que se trabaja (ver capítulo 5).

Dentro de esta interfaz de diálogos orales se ha comprobado que uno de los aspectos primordiales para conseguir una interacción satisfactoria se halla en el empleo del contexto físico del entorno. La interfaz oral permite establecer de forma automática diálogos de clarificación específicos para el entorno en que se encuentra y su contexto actual. Por lo tanto, las mismas interacciones tendrán interpretaciones, acciones y respuestas diferentes para contextos distintos y entornos dispares.

Además, el gestor de diálogos permite recuperarse de algunos de los errores más comunes en este tipo de sistemas. En el caso de que se produzca un error de reconocimiento, una elipsis o el usuario haya proporcionado menos información de la necesaria el sistema utiliza el contexto real del entorno para intentar recuperarse de estas situaciones. Todo esto se realiza empleando la capa *middleware* descrita anteriormente.

La última de las aportaciones presentada consiste una evaluación del sistema de diálogos automáticos dentro del entorno inteligente desarrollado por 37 usuarios no entrenados ni familiarizados con entornos de interacción oral (ver capítulo 6). Los resultados demuestran el alto rendimiento del sistema basándose en la información de contexto y la aceptación de las personas que lo han utilizado. De entre las oraciones recibidas en donde quedaba omitida parte de la información necesaria, en un 92% de las ocasiones la interfaz ha sido capaz de realizar una interpretación parcial o total correcta que permitía continuar con el diálogo. Además, mediante el cuestionario realizado se comprueba que existe un 85% de satisfacción de los usuarios en el empleo de la interfaz presentada dentro del entorno real. Asimismo, se han realizado pruebas de escalabilidad de la interfaz de diálogos orales. El sistema de diálogos desarrollado puede trabajar con entornos de tamaño medio-grande con presencia de hasta aproximadamente 100 entidades diferentes.

7.2 Publicaciones a las que ha dado lugar este trabajo

El trabajo aquí presentado ha expuesto sus ideas iniciales en foros de investigación nacionales e internacionales. Entre ellos se encuentran revistas, capítulos de libros y conferencias.

7.2.1 Revistas internacionales

- Germán Montoro, Pablo A. Haya and Xavier Alamán. 2004. "Context adaptive interaction with an automatically created spoken interface for intelligent environments". The 2004 IFIP International Conference on Intelligence in Communication Systems (INTELLCOMM 04). Bangkok, Thailand. November 23-26, 2004. *Lecture Notes in Computer Science (LNCS)*, volume number 3283. ISSN 0302-9743.
- Pablo A. Haya, Germán Montoro and Xavier Alamán. 2004. "A prototype of a context-based architecture for intelligent home environments". International Conference on Cooperative Information Systems (CoopIS 2004), Larnaca, Cyprus. October 25-29, 2004. *Lecture Notes in Computer Science (LNCS)*, volume number 3290. ISSN 0302-9743
- Germán Montoro, Xavier Alamán and Pablo A. Haya. 2004. "A plug and play spoken dialogue interface for smart environments". Fifth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing'04). Seoul, Korea. February 15-21, 2004. *Lecture Notes in Computer Science (LNCS)*, volume number 2945. ISSN 0302-9743.
- Pablo Haya, Xavier Alamán and Germán Montoro. 2001. "A Comparative Study of Communication Infrastructures for the Implementation of Ubiquitous Computing". *UPGRADE, The European Journal for the Informatics Professional*, 2, 5. ISSN 1684-5285.

7.2.2 Capítulos de libro

- Germán Montoro, Xavier Alamán and Pablo A. Haya. 2004. "Spoken interaction in intelligent environments: a working system". *Advances in Pervasive Computing*. Alois Ferscha, Horst Hoertner and Gabriele Kotsis, Eds. Austrian Computer Society (OCG). ISBN 3-85403-176-9.

7.2.3 Conferencias internacionales

- Pablo A. Haya, Germán Montoro, Xavier Alamán, Rubén Cabello and Javier Martínez. 2004. "Extending an XML environment definition language for spoken dialogue and web-based interfaces". Developing User Interfaces with XML: Advances on User Interface Description Languages Workshop at AVI04. Gallipoli, Italy. May 25, 2004.
- Germán Montoro, Xavier Alamán and Pablo A. Haya. 2003. "Facts and challenges in a dialogue system for Smart Environments". 3rd Workshop on Knowledge and Reasoning in Practical Dialog Systems. 18th International Joint Conference on Artificial Intelligence (IJCAI'03). Acapulco, Mexico. August 9-15, 2003.

- Xavier Alamán, Rubén Cabello, Francisco Gómez-Arriba, Pablo Haya, Antonio Martínez, Javier Martínez, Germán Montoro. 2003. "Using context information to generate dynamic user interfaces". 10th International Conference on Human-Computer Interaction, HCI International 2003. Crete, Greece. June 22-27, 2003.

7.2.4 Conferencias nacionales

- Germán Montoro, Pablo A. Haya and Xavier Alamán. 2004. "Interacción adaptada al contexto en una interfaz de diálogos orales para entornos inteligentes". III Jornadas en Tecnología del Habla. Valencia. 17-19 de noviembre, 2004.
- Germán Montoro, Xavier Alamán and Pablo A. Haya. 2004. "Interacción con entornos inteligentes mediante diálogos basados en contexto". V Congreso Interacción Persona Ordenador (Interacción 2004). Lleida. 3-7 de mayo, 2004.
- Pablo Haya, Xavier Alamán y Germán Montoro. 2003. "El proyecto Interact: el rol de la información contextual". IV Congreso Interacción Persona Ordenador (Interacción 2003). Vigo. 11-13 de junio, 2003.
- Xavier Alamán, Pablo Haya and Germán Montoro. 2001. "El proyecto InterAct: Una arquitectura de pizarra para la implementación de Entornos Activos". II Congreso Interacción Persona Ordenador (Interacción 2001). Salamanca. 16-18 de mayo, 2001.
- Xavier Alamán, Pablo Haya and Germán Montoro. 2000. "ODISEA: Hacia un entorno inteligente basado en un interfaz en lenguaje natural". I Jornadas en Tecnología del Habla. Sevilla. 6-10 de noviembre, 2000.

7.3 Trabajo futuro

Tanto la línea de investigación sobre entornos inteligentes (ver apartado 2.4) como la de interfaces de diálogos orales (ver apartado 2.9) permanecen en la actualidad abiertas con diversos grupos de investigación trabajando activamente en ellas. Muchos son los retos que todavía se deben afrontar en ambos campos por separado o, como se ha mostrado en este trabajo, en conjunto.

La investigación aquí presentada es el embrión de una línea que pretende asegurar su permanencia en el futuro, explorando nuevas técnicas y posibilidades y ahondando y mejorando algunas de las ideas propuestas.

El reconocimiento de voz ha sido explícitamente omitido en esta investigación (ver apartado 3.9.1.1). Sin embargo su buen rendimiento constituye un punto necesario para el desarrollo de este tipo de interfaces (como se especifica en las propuestas realizadas en el cuestionario de evaluación, ver apartado 6.2.2). Una de las posibilidades que se pueden explorar consiste en adaptar el reconocimiento al entorno sobre el que se trabaja. Por ejemplo, en el entorno del hogar es muy fácil conocer quiénes serán los usuarios de la interfaz en la mayoría de las ocasiones. El sistema podría entrenarse y

adaptarse a estos usuarios, mejorando considerablemente su rendimiento. Por supuesto, estas características dependen de la naturaleza particular de cada espacio. Dentro de un entorno laboral una aproximación de este estilo sería fácil de desarrollar dentro un despacho pero mucho más difícil en una sala de reuniones.

Esta idea enlaza con la posibilidad de conocer a los ocupantes del entorno y al usuario de la interfaz en cada momento (la primera de ellas ya se realiza en el entorno desarrollado, aunque todavía de forma muy básica, la segunda aún no se ha contemplado). Esta información puede permitir adaptar la interfaz a sus usuarios, personalizándola según los casos. Esta es, además, otra de las propuestas manifestadas en el cuestionario de evaluación.

La personalización de la interfaz también podría aplicarse al histórico de acciones (ver apartado 5.3.3.1). Actualmente, este histórico se emplea fundamentalmente para la resolución de la anáfora pronominal (ver apartado 5.3.3.8). Un histórico individualizado para cada usuario permitiría al sistema adaptarse mejor a cada uno de ellos, permitiendo a la interfaz reaccionar de forma proactiva (otra de las solicitudes expuestas en la evaluación).

Una de las premisas básicas durante la investigación y desarrollo llevados a cabo ha sido que sus resultados se debían traducir en entornos de interacción reales. Por un lado, este aspecto limita su envergadura ya que se debe estudiar y desarrollar la capa física de interacción para cada entidad del entorno pero, por otro lado, proporciona resultados tangibles que permiten obtener una valoración real de las ideas desarrolladas. Por esto, se va a seguir ampliando el número de entidades con las que se puede interaccionar. Como resultado, se espera conseguir entornos más complejos pudiendo establecer divisiones entre varios espacios, aproximándose así en mayor medida al paradigma de un entorno cotidiano. Sobre este nuevo o nuevos entornos con un mayor número de posibilidades de interacción se deberían hacer nuevas pruebas de evaluación, con el fin de corroborar o modificar los datos obtenidos hasta el momento.

Una de las ideas que siempre se ha planteado en el desarrollo de esta investigación es la posibilidad de implementar estos entornos en espacios de ocupación permanentes (como despachos o salas de reunión) con las finalidades de obtener resultados de evaluación continuos, comprobar el interés de los usuarios en su uso y conseguir información en espacios plenamente reales.

Un aspecto importante en el desarrollo del presente trabajo ha sido el uso de un lenguaje de descripción del entorno y sus interfaces. Este lenguaje se debería aumentar y mejorar para poder definir múltiples interfaces mediante una descripción unitaria. Este podría ser también un paso previo para proporcionarle mayor multimodalidad ofrecer mayor soporte en la interacción a las diferentes interfaces que lo componen.

Entre las nuevas modalidades de interacción que se podrían incorporar al sistema se encuentra un módulo de reconocimiento de gestos que permita actuar junto con la interfaz de diálogos orales para proporcionar una interacción más natural con el

entorno. Otras posibilidades de interacción se encuentran en paneles y pizarras táctiles y digitales, así como el uso de PDAs y teléfonos móviles.

Bibliografía y referencias

"No hay libro tan malo que no tenga algo bueno"
Bachiller Sansón Carrasco – Capítulo III – Segunda Parte

"La pluma es lengua del alma"
Don Quijote – Capítulo XVI – Segunda Parte

ABOWD, G.D. 1998. Software design issues for ubiquitous computing. In Proceedings of IWV'98.

ABOWD, G.D.; ATKESON, C.; BROTHERTON, J.; ENQVIST, T.; GULLEY, P. and LEMON, J. 1998. Investigating the capture, integration and access problem of ubiquitous computing in an educational setting. In Proceedings of CHI'98 (Los Angeles, GA, April 18-23), 440-447.

ABOWD, G.D.; ATKESON, C. and ESSA I. 1997. Computational perception in future computing environments. In Proceedings of PUI'97.

ABOWD, G.D.; ATKESON, C.; FEINSTEIN, A.; GOOLAMABBAS, Y.; HMELO, C.; REGISTER, S.; SAWHNEY, N. and TANI, M. 1996. Classroom 2000: Enhancing classroom interaction and review. Technical Report GIT-GVU-96-21, GVU Center, Georgia Institute of Technology.

ADLER, A. and DAVIS, R. 2004. Speech and Sketching for Multimodal Design. In Proceedings of the 9th International Conference on Intelligent User Interfaces.

ALEXANDERSSON, J. 1996. Some Ideas for the Automatic Acquisition of Dialogue Structure. In Proceedings of the 11th Twente Workshop on Language Technology.

ALLEN, J.F.; SCHUBERT, L.K.; FERGUSON, G.; HEEMAN, P.; HWANG, G.H.; KATO, T.; LIGHT, M.; MARTIN, N.G.; MILLER, B.W.; POESIO, M. and TRAUM, D.R. 1995. The TRAINS Project: A case study in building a conversational planning agent. Journal of Experimental and Theoretical AI, 7.

ALLEN, J.F.; MILLER, B.W.; RINGGER, E.K. and SIKORSKI, T. 1996. A Robust System for Natural Spoken Dialogue. In Proceedings of the 1996 Annual Meeting of the Association for Computational Linguistics.

ALM, N.; ARNOTT, J.L. and NEWELL, A.F. 1992. Prediction and conversational momentum in an augmentative communication system. Communications of the ACM, 35, 5 (May 1992), 46-57.

AMORES, J.G.; QUESADA, J.F. 2001. Dialogue Moves for Natural Command Languages. Procesamiento del Lenguaje Natural, 27, 81-88.

ANDRY, F. and THORNTON, S. 1991. A parser for speech lattices using a UCG grammar. In Proceedings of the 2nd European Conference on Speech Communication and Technology.

ARONS, B. and MYNATT, E.D. 1994. The future of speech and audio in the interface. In Proceedings of CHI'94 (Boston, April 24-28), 465.

AUSTIN, J.L. 1962. How to do things with words. Harvard University Press, Boston.

BAPTIST, L. and SENEFF, S. 2000. Genesis-II: A Versatile System for Language Generation in Conversational System Applications. In Proceedings of ICSLP.

BENET, B. 1995. Dictation Systems for Windows: Dragon, IBM, Kurzweil. Seybold Report on Desktop Publishing, 9, 10 (June 1995), 12-19.

BOBICK, A.; INTILLE, S.; DAVIS, J.; BAIRD, F.; PINHANEZ, C.; CAMPBELL, L.; IVANOV, Y.; SCHÜTTE, A. and WILSON, A. 1997 The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment, Technical report 398, MIT Media Laboratory, Perceptual Computing Section.

BJÖRK, S. 1998. Making guides entertaining. In Proceedings of ECAI Workshop on AI and Entertaiment (Brighton, UK).

BRANDSTEIN, M.S.; ADCOCK, J.E. and SILVERMAN, H.F. 1997. A closed-form location estimator for use with room environment microphone arrays. IEEE Transactions on Speech and Audio Processing, 5, 1, 45-50.

BRETAN, I.; EREBACK, A.L.; MACDERMID, C. and WAERN, A. 1995. Simulation-based dialogue design for speech-controlled telephone services. In Proceedings of CHI'95 (Denver, CO, May 7-11).

BROTHERTON, J.A. and ABOWD, G.D. 1998. Rooms take note: Room takes notes! In Proceedings of the AAAI Spring Symposium on Intelligent Environments (AAAI98).

BRUMITT, B. and CADIZ, J.J. 2001. Let There Be Light! Comparing Interfaces for Homes of the Future, In Proceedings of the 2001 IFIP TC.13 Conference on Human Computer Interaction (Interact 2001).

BRUMITT, B. and SHAFER, S.A.N. 2001. Better Living Through Geometry. Springer-Verlag Journal on Ubiquitous Computing, 5, 1.

BRUMITT, B.; MEYERS, B.; KRUMM, J.; KERN, A. and SHAFER, S.A.N. 2000. EasyLiving: Technologies for Intelligent Environments. In Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing, Bristol, UK.

CARMONA, J.; CERVELL, S.; ATSERIAS, J.; CERVELL S.; MÁRQUEZ, L.; MARTÍ, M.A.; PADRÓ, L.; PLACER, R.; RODRÍGUEZ, H.; TAULÉ, M. and TURMO, J. 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).

CHAPMAN, D. 1992. Computer rules, conversational rules. Association for Computational Linguistics, 18, 4, 531-536.

CHEN, M. 2001. Design of a Virtual Auditorium. In Proceedings of ACM Multimedia 2001.

CHU-CARROLL, J. and BROWN, M.K. 1997. Tracking initiative in collaborative dialogue interactions. In Proceedings of the 35th annual meeting of the ACL.

CHURCHER, G.E.; ATWELL, E.S. and SOUTER, C. 1997. Dialogue management systems: a survey and overview. Report 97.6, School of Computer Studies, University of Leeds.

CLARK, H.H. and BRENNAN, S.E. 1991. Grounding in communication. Shared Cognition; Thinking as Social Practice, J. Levine, L.B. Resnick, and S.D. Behrand, Eds., APA Books, Washington, 127-149.

COEN, M.H. 1997. Building brains for rooms: Designing distributed software agents. In Proceedings of IAAI97.

COEN, M.H. 1998. Design Principles for Intelligent Environments. In Proceedings of the AAAI Spring Symposium on Intelligent Environments (AAAI98).

COEN, M.H. 1999. The future of human-computer interaction or How I learned to stop worrying and love my Intelligent Room. In IEEE Intelligent Systems, 14, 2.

COEN, M.H.; PHILLIPS, B.; WASHAWSKY, N.; WEISMAN, L.; PETERS, S. and FININ, P. 1999a. Meeting the computational needs of intelligent environments: The MetagGue System. In Proceedings of 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99, Dublin, Dec 1999), P. Nixon, G. Lacey and S. Dobson, Eds. Springer-Verlag, London, 201-212.

COEN, M.H.; WEISMAN, L.; THOMAS, K. and GROH, M. 1999b. A context sensitive natural language modality for an intelligent room. In Proceedings of 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99, Dublin, Dec 1999), P. Nixon, G. Lacey and S. Dobson, Eds. Springer-Verlag, London, 68-79.

COOK, D. J. and YOUNGBLOOD, M. 2004. Living in an Intelligent Environment. Ergonomics in Design, 2004.

COOK, D. J.; YOUNGBLOOD, M.; HEIERMAN, E.; GOPALRATNAM, K.; RAO, S.; LITVIN, A. and KHAWAJA, F. 2003. MavHome: An Agent-Based Smart Home. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications

COOPER, R. 1997 Information states, attitudes and dialogue. In Proceedings of the Second Tblisi Symposium on Language, Logic and Computation.

COOPERSTOCK, J.R.; TANIKOSHI, K.; BEIRNE, G.; NARINE, T. and BUXTON, W. 1995. Evolution of a reactive environment. In Proceedings of CHI'95 (Denver, CO, May 7-11).

DAHLBÄCK, N. and JÖNSSON, A. 1992. An empirically based computationally tractable dialogue model. In Proceedings of the 14th Annual Conference of the Cognitive Science Society (COGSCI'92, July 1992).

DAHLBÄCK, N.; JÖNSSON, A. and AHRENBERG, L. 1993. Wizard of Oz studies: Why and how. In Proceedings of ACM IUI'93 (Orlando, FL, Jan 4-7), 193-200.

DAVIS, E. 1990. Representations of Commonsense Knowledge. Morgan Kaufmann Publishers.

ELTING, C.; RAPP, S.; MÖHLER, G. and STRUBE, M. 2003. Architecture and Implementation of Multimodal Plug and Play. In Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI-PUI '03), Vancouver, B.C., Canada.

ENGELMORE, R. and MORGAN, T. 1988. Blackboard Systems. Addison-Wesley.

ESSA, I.A. 1999. Ubiquitous sensing for smart and aware environments: Technologies towards the building of an Aware Home. Position Paper for the DARPA/NSF/NIST Workshop on Smart Environments.

FEINER, S.; MACINTYER, B. and SELIGMANN, D. 1993. Knowledge-based augmented reality. Communications of the ACM, 36, 7, 53-62.

FERGUSON, G. and ALLEN, J.F. 1998. TRIPS: An Intelligent Integrated Problem-Solving Assistant. In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98).

FININ, T. FRITZSON, R.; MCKAY, D. and MCENTIRE, R. 1994. KQML as an agent communication language. In Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94).

GAJOS, K.; FOX, H. and SHROBE, H. 2002. End User Empowerment in Human Centered Pervasive Computing. Pervasive 2002, Zurich, Switzerland.

GARAY-VICTORIA, N. and GONZÁLEZ-ABASCAL, J. 1997. In Proceedings of ACM IUI'97 (Orlando, FL, Jan 6-9), 241-244.

GAVER, W.W. 1991. Sound support for collaboration. In Proceedings of the 2nd European Conference on Computer-Supported Cooperative Work (ECSCW'91, Amsterdam, The Netherlands, Sep 25-27), 355-362.

GEUTNER, P.; DENECKE, M.; MEIER, U.; WESTPHAL, M. and WAIBEL, A. 1998. Conversational Speech Systems for On-Board Navigation and Assistance. In Proceedings of the 1998 International Conference on Spoken Language Processing (ICSLP'98, Sydney, Australia).

GINZBURG, J. 1996. Interrogatives: Questions, Facts and Dialogue. The Handbook of Contemporary Semantic Theory.

GLASS, J.; CHANG, J. AND MCCANDLESS, M. 1996. A Probabilistic Framework for Feature-Based Speech Recognition. In Proceedings of ICSLP 96.

GÓMEZ, P.; ÁLVAREZ, A.; MARTÍNEZ, R.; RODELLAR, V. and NIETO, V. 1998. A noise-robust speech processing and recognition development system. Technologies for the information society: Developments and opportunities, J.-Y. Roger et al., Eds. IOS Press, 803-810.

GÖRZ, G.; SPILKER, J.; STROM, V. and WEBER, H. 1999. Architectural Considerations for Conversational Systems-The Verbmobil/INTARC Experience. In Proceedings of the First International Workshop on Human Computer Conversation.

GRICE, H. 1975. Logic and conversation. Syntax and Semantics: Speech Acts, Cole and Morgan, Eds., Academic Press.

GUENTCHEV, K.Y. and WENG, J.J. 1998. Learning-based three dimensional sound localization using a compact non-coplanar array of microphones. In Proceedings of the AAAI Spring Symposium on Intelligent Environments (AAAI98).

GUIMBRÈTIERE, F. and WINOGRAD, T. 2000. FlowMenu: Combining Command, Text, and Data Entry. 2000. In Proceedings of ACM Symposium on User Interface Software and Technology (UIST).

GUINN, C.I. 1999. Evaluating mixed-initiative dialog. IEEE Intelligent Systems, 14, 5, 21-23.

HANSSENS, N.; KULKARNI, A.; TUCHINDA, R. and HORTON, T. 2002. Building Agent-Based Intelligent Workspaces. In ABA Conference Proceedings.

HANSEN, B.; NOVICK, D.G. and SUTTON, S. 1996. Systematic design of spoken prompts. In Proceedings of CHI'96 (Vancouver, April 13-18).

HORVITZ, E. and PAEK, T. 1999. A Computational Architecture for Conversation. In Proceedings of the Seventh International Conference on User Modelling, Banff, Canada.

INTILLE, S.S. 2002. Designing a Home of the Future. IEEE Pervasive Computing, April-June 2002, 80-86.

INTILLE, S.S. and LARSON, K. 2003. Designing and Evaluating Supportive Technology for Homes. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics.

INTILLE, S.S. LEE, V. and PINHANEZ, C. 2003. Ubiquitous Computing in the Living Room: Concept Sketches and an Implementation of a Persistent User Interface. In Proceedings of UBICOMP 2003 Video Program.

INTILLE, S.S.; MUNGUIA TAPIA, E.; RONDONI, J.; BEAUDIN, J.; KUKLA, C.; AGARWAL, S. and Bao, L. 2003. Tools for Studying Behavior and Technology in Natural Settings. In Proceedings of UBICOMP 2003.

ISHII, H. and ULLMER, B. 1997. Tangible bits: Towards seamless interfaces between people, bits and atoms. In Proceedings of CHI'97 (Atlanta, GA, March 22-27), 234-241.

JOHANSON, B.; FOX, A. and WINOGRAD, T. 2002. The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. IEEE Pervasive Computing Magazine, 1(2).

JOHANSON, B.; KICIMAN, E. and FOX, A. 2000. Moving Data and Interfaces in an Interactive Workspace. Handheld and Ubiquitous Computing (HUC2K) Workshop on Infrastructure for Smart Devices, Bristol, England.

KARAT, C.; HALVERSON, C.; HORN, D. and KARAT, J. 1999. Patterns of entry and correction in large vocabulary continuous speech recognition systems. In Proceedings of CHI'99 (Pittsburgh, PA, May 15-20), 568-575.

KIDD, C.D.; ORR, R.; ABOWD, G.D.; ATKENSON, C.G.; ESSA, I.A.; MACINTYRE, B.; MYNATT, E.; STARNER, T.E. and NEWSTETTER, W. 1999. The Aware Home: A Living Laboratory for Ubiquitous Computing Research. In Proceedings of the Second International Workshop on Cooperative Buildings (CoBuild'99).

KNIGHT, S.; GORRELL, G.; RAYNER, M.; MILWARD, D.; KOELING, R. and LEWIN, I. 2001. Comparing grammar-based and robust approaches to speech understanding: a case study. In Proceedings of Eurospeech 2001.

KOWTKO, J.; ISARD, S. and DOHERTY, G.M. 1992. Conversational Games Within Dialogue. Research Paper HCRC/RP-31, Human Communications Research Centre, University of Edinburgh.

KRUMM, J.; HARRIS, S.;MEYERS, B.; BRUMITT, B.; HALE, M. and SHAFER S.A.N. 2000. Multi-camera multi-person tracking for EasyLiving. In Proceedings of IEEE Intl. Workshop on Visual Surveillance, Dublin, Ireland.

KUHN, T.; NIEMANN, H.; SCHUKAT-TALAMAZZINI, E.G.; ECKERT, W. and RIECK, S. 1992. Context-Dependent Modeling in a Two-Stage HMM Word Recognizer for Continuous Speech. In Signal Processing VI: Theories and Applications, 1.

LAI, J. and VERGO, J. 1997. Medspeak: report creation with continuous speech recognition. In Proceedings of CHI'97 (Atlanta, GA, March 22-27), 431-438.

LARSSON, L. AND TRAUM, D. 2000. Information state and dialogue management in the TRINDI Dialogue Move Engine Toolkit. Natural Language Engineering special issue on Best Practice in Dialogue Systems Design.

LE GAL, C.; MARTIN, J.; LUX, A. and CROWLEY, J.L. 2001. Smart Office: Design of an Intelligent Environment. IEEE Intelligent Systems, 16, 4.

LEHNERT, W.G. 1977. A conceptual theory of question answering. In Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI-77, Cambridge, Massachusetts), 158-164.

LEVINSON, S.C. 1981. Some pre-observations on the modelling of dialogue. Discourse Processes, 4.

LIEBERMAN, H.; NARDI, B.A. and WRIGHT, D. 1998. Grammex: Defining grammars by example. In Proceedings of CHI'98 (Los Angeles, GA, April 18-23), 11-12.

LLEIDA, E.; FERNÁNDEZ, J. and MASGRAU E. 1998. Robust Continuous Speech Recognition System based on a Microphone Array. In Proceedings of ICASSP-98, 1, 241–244.

MACKAY, W.E.; PAGANI, D.S.; FABER L.; INWOOD, B.; LAUNIAINEN, P., BRENTA, L. and POUZOL, V. 1995. Ariel: Augmenting paper engineering drawings. In Proceedings of CHI'95 (Denver, CO, May 7-11).

MANKOFF, J. and ABOWD, G.D. 1997. Domisilica: Providing ubiquitous access to the home. Technical Report GIT-GVU-97-17, GVU Center, Georgia Institute of Technology.

MANKOFF, J. and SCHILIT, B. Supporting knowledge workers beyond the desktop with Palplates. In Proceedings of CHI'97 (Atlanta, GA, March 22-27).

MANKOFF, J. SOMERS, J. and ABOWD, G.D. 1998. Bringing people and places together. In Proceedings of the AAAI Spring Symposium on Intelligent Environments (AAAI98).

MANSE 1999. 1st International Workshop on Managing Interactions in Smart Environments (Trinity College, Dublin, Decr 13-14 1999).

MARTÍ, M.A.; RODRÍGUEZ, H. and SERRANO, J. 1998 Declaración de categorías morfosintácticas. Doc.ITEM n. 2, Universitat Politècnica de Catalunya and Universitat de Barcelona.

MARTIN, P.; CRABBE, F.; ADAMS, S.; BAATZ, E. and YANKELOVICH, N. 1996. SpeechActs: A spoken-language framework. Computer, IEEE Computer Society, 29, 7, 33-40.

MARTÍNEZ, A.E., CABELLO, R., GÓMEZ, F. J. and MARTÍNEZ, J. INTERACT-DM. 2003. A Solution For The Integration Of Domestic Devices On Network Management Platforms. IFIP/IEEE International Symposium on Integrated Network Management. Colorado Springs, USA.

MARX, M. and SCHMANDT, C. 1996. MailCall: Message presentation and navigation in a nonvisual environment. In Proceedings of CHI'96 (Vancouver, April 13-18).

MATEAS, M.; SALVADOR, T.; SCHOLTZ, J. and SORENSEN, D. 1996. Engineering Ethnography in the Home. CHI 96 Conference Companion.

MCGLASHAN, S.; FRASER, N.M.; GILBERT, N.; BILANGE, E.; HEISTERKAMP, P. and YOUD, N.J. 1992. Dialogue management for telephone information services. In Proceedings of the 3rd International Conference on Applied Natural Language Processing.

MCLAUGHLIN, A.C.; ROGERS, W.A. and FISK, A.D. 2003. Effectiveness of audio and visual training presentation modes for glucometer calibration. Human Factors and Ergonomics Society 47th Annual Meeting

MCTEAR, M.F. 2002. Spoken dialogue technology: enabling the conversational interface. *ACM Computing Surveys* 34 , 1.

MILWARD, D. and BEVERIDGE, M. 2003. Ontology-based dialogue systems. IJCAI WS on Knowledge and reasoning in practical dialogue systems, Acapulco, Mexico.

MONTORO, G. 2000. Entornos inteligentes. Un nuevo paradigma de interacción. Trabajo de estudios avanzados. Universidad Autónoma de Madrid.

MOZER, M.C. 1998. The neural network house: An environment that adapts to its inhabitants. In Proceedings of the AAAI Spring Symposium on Intelligent Environments (AAAI98).

MOZER, M.C. 1999. An intelligent environment must be adaptative. IEEE Intelligent Systems, 14, 2, 11-13.

MOZER, M. C. 2004. Lessons from an adaptive house. Smart environments: Technologies, protocols, and applications, D. Cook & R. Das (Eds.). J. Wiley & Sons.

MOZER, M.C.; DODIER, R.H.; ANDERSON, M.; VIDMAR, L.; CRUICKSHNAK III, R.F. and MILLER, D. 1995. The Neural Network House: An overview. Current trends in connectionism, L. Niklasson and M. Boden, Eds., Lawrence Erlbaum, Hillsdale, NJ, 371-380.

MUNGUIA TAPIA, E. INTILLE, S.S. and LARSON, K. 2004. Activity recognition in the home using simple and ubiquitous sensors. In Proceedings of International Conference Pervasvie 2004, Viena, Austria.

MYNATT, E.D.; BACK, M.; WANT, R. and FREDERICK, R. 1997. Audio Aura: Light-weight audio augmented reality. In Proceedings of ACM UIST'97 (Banff, Canada), 211-212.

NAGAO, K. and REKIMOTO J. 1995. Ubiquitous talker: Spoken language interaction with real world objects. In Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95), Vol. 2, 1284-1290.

NIXON, P.; DOBSON, S. and LACEY, G. 1999. Smart Environments: some challenges for the computing community. In Proceedings of 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99, Dublin, Dec 1999), P. Nixon, G. Lacey and S. Dobson, Eds. Springer-Verlag, London, 1-4.

OVIATT, S.; DEANGELI, A. and KUHN, K. 1997. Integration and sychronization of input modes during multimodal human-computer interaction. In Proceedings of CHI'97 (Atlanta, GA, March 22-27), 415-422.

PARKINSON, R.C.; COLBY, K.M. and FAUGHT, W. 1977. Conversational language comprehension using integrated pattern-matching and parsing. Artificial Intelligence, 9, 111-134.

PECKHAM, J. 1991. Speech understanding and dialogue over the telephone: an overview of progress in the sundial project. In Proceedings of the 2nd European Conference on Speech Communication and Technology.

PECKHAM, J. 1993. A new generation of spoken dialogue systems: results and lessons from the SUNDIAL project. In Proceedings of the 3rd European Conference on Speech Communication and Technology.

PENTLAND, A. 1996. Smart rooms. Scientific American, 274, 4, 68-76.

PFLEGER, P.; ALEXANDERSSON, J. and BECKER, T. 2003. A robust and generic discourse model for multimodal dialogue. In Proceedings of the 3rd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Acapulco, Mexico.

PINHAEZ, C. 2001. The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces. In Proceedings of Ubicomp 2001, Atlanta, Georgia.

PONNEKANTI, S.R.; ROBLES, L.A. and FOX, A. 2002. User Interfaces for Network Services: What, from Where, and How. In Proceedings of 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002), Callicoon, NY.

PULLUM, G.K. and GAZDAR, G. 1982. Natural Languages and Context-Free Languages. Linguistics and Philosophy, 4.

QUESADA, J.F. and AMORES, J.G. 2002. Knowledge-based reference resolution for dialogue management in a home domain environment. In Proceedings of the sixth workshop on the semantics and pragmatics of dialogue.

QUESADA, J.F.; GARCÍA, F., SENA, E.; BERNAL, J.A. and AMORES, G. 2001. Dialogue management in a home machine environment: Linguistic components over an agent architecture. In SEPLN. 27, 89-98.

RAYNER, M.; LEWIN, I.; GORRELL, G. and BOYE, J. 2001. Plug and Play Speech Understanding. In Proceedings of 2nd SIGdial Workshop on Discourse and Dialogue.

REITHINGER, N.; ALEXANDERSSON, J.; BECKER, T.; BLOCHER, A.; ENGEL, R.; LÖCKELT, M.; MÜLLER, J.; PFLEGER, N.; POLLER, P.; STREIT, M. and TSCHERNOMAS, V. 2003. SmartKom - Adaptive and Flexible Multimodal Access to Multiple Applications. In Proceedings of ICMI 2003.

RODIN, J. and LANGER, E.J. 1977. Long-term effects of a control-relevant intervention with the institutionalized aged. Journal on Personality and Social Psychology, 35, 12.

ROSU, M.; SCHWAN, K. and FUJIMOTO, R. 1997. Supporting Parallel Applications on Clusters of Workstations: The Intelligent Network Interface Approach. In Proceedings of the 6th IEEE International Symposium on High Performance Distributed Computing, (HPDC '97, Portland, OR).

ROY, D. and PENTLAND, A. 1998a. Learning words from natural audio-visual input. In Proceedings of the International Conference of Spoken Language Processing (Sydney, 1998), Vol. 4, 1279.

SCHANK, R. and ABELSON, R. 1977. Scripts, Plans and Goals. Erlbaum, Hillsdale, New Jersey.

SCHMANDT, C. 1994. Voice communication with computers: conversational systems. Van Nostrand Reinhold, New York.

SEARLE, J. R. 1969. Speech Acts. An Essay in the Philosophy of Language. Cambridge University Press.

SEARLE, J.R. 1976. A classification of illocutionary acts. Language in Society, 5.

SENEFF, S. 1992. TINA: A natural language system for spoken language applications. Computational Linguistics, 18, 1.

SENEFF, S.; HURLEY, E.; LAU, R.; PAO, C.; SCHMID, P. and ZUE, V. 1998. GALAXY-II: A Reference Architecture for Conversational System Development. In Proceedings of ICSLP 98, Sydney, Australia.

SENEFF, S. AND POLIFRONI, J. 2000. Dialogue Management in the Mercury Flight Reservation System. ANALP-NAACL Workshop on Conversational Systems.

SHAFER, S.A.N. 1999. Ten dimensions of ubiquitous computing. In Proceedings of 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99, Dublin, Dec 1999), P. Nixon, G. Lacey and S. Dobson, Eds. Springer-Verlag, London, 5-16.

SHAFER, S.A.N.; BRUMITT, B. and CADIZ, JJ. 2001. Interaction Issues in Context-Aware Interactive Environments. Special issue on context-aware computing. Human-Computer Interaction Journal, 16, 2-4.

SHAFER, S.A.N. 1999. Ten dimensions of ubiquitous computing. In Proceedings of 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99, Dublin, Dec 1999), P. Nixon, G. Lacey and S. Dobson, Eds. Springer-Verlag, London, 5-16.

SHAFER, S.A.N.; KRUMM, J.; BRUMITT, B.; MEYERS, B.; CZERWINSKI, M. and ROBBINS, D. 1998. The new easyliving project at microsoft research. In Proceedings of DARPA/NIST Smart Spaces Workshop, Gaithersburg, Maryland, July 30-31.

SIMS, D. 1994. New realities in aircraft design and manufacture. IEEE Computer Graphics and Applications, 14, 2, 91.

STAFFORD-FRASER, Q. and ROBINSON, P. 1996. BrightBoard: A video-augmented environment. In Proceedings of CHI'96 (Vancouver, April 13-18).

STILLMAN, S. and ESSA, I. 2001. Towards Reliable Multimodal Sensing in Aware Environments. Perceptual User Interfaces (PUI 2001) Workshop.

SUTTON, S. and COLE, R. 1997. The CSLU Toolkit: Rapid prototyping of spoken language systems. In Proceedings of ACM UIST'98 (Banff Canada, Oct 14-17), 85-86.

TETZLAFF, L.; KIM, M. and SCHLOSS, R.J. 1995. Home Health Care Support. CHI 95 Conference Companion.

TORRANCE, M.C. 1995. Advances in Human-Computer Interaction: The Intelligent Room. In Proceedings of CHI'95 (Denver, CO, May 7-11).

TRAN-HUY, K. 1998. MELISSA methods&tools for natural language interfacing to standard software. Technologies for the information society: Developments and opportunities, J.-Y. Roger et al., Eds. IOS Press, 583-590.

WAHLSTER, W.; REITHINGER, N. and BLOCHER, A. 2001. SmartKom: Multimodal Communication with a Life-Like Character. In Proceedings of Eurospeech2001, Aalborg, Denmark.

WALKER, M.A.; FROMER, J.; DI FABBRIZIO, G.; MESTEL, C. And HINDLE D. 1998. What can I say?: Evaluating a spoken language interface to email. In Proceedings of CHI'98 (Los Angeles, GA, April 18-23), 582-589.

WALKER, M.A.; LITMAN, D.J.; KAMM, C.A. and ABELLA, A. 1997. PARADISE: A framework for evaluating spoken dialogue agents. In Proceedings of the 35th annual meeting of the ACL.

WANT, R.; HOPPER, A.; FALCAO, V. and GIBBONS, H. 1992. The active badge location system. ACM Transactions on Information Systems, 10, 1, 91-102.

WANT, R.; SCHILIT, B.N.; ADAMS, N.I.; GOLD, R.; PETERSEN, K.; GOLDEBERG, D.; ELLIS, J.R. and WEISER, M. 1995. The PARCTAB ubiquitous computing experiment. Technical report CSL-05-1, Xerox Palo Alto Research Center.

WARD, K. and NOVICK, D.G. 1995. Integrating multiple cues for spoken language understanding. In Proceedings of CHI'95 (Denver, CO, May 7-11).

WEIZENBAUM, J. 1966. ELIZA. A computer program for the study of natural language communication between man and machine. Communications of the ACM, 9, 1, 36-45.

WELLNER, P. 1993. Interacting with paper on the digital desk. Communications of the ACM, 36, 7, 86-96.

WEISER, M. 1991. The computer of the 21st century. Scientific American, 265, 3, 66-75.

WEISER, M. 1994. The world is not a desktop. ACM Interactions, 1, 1, 7-8.

WOLF, C.G. and ZADROZNY, W. 1998. Evolution of the conversation machine: a case study of bringing advanced technology to the marketplace. In Proceedings of CHI'98 (Los Angeles, CA, April 18-23), 488-495.

YAN, H. and SELKER, T. 2000. Context-aware office assistant. In Proceedings of ACM IUI'2000 (New Orleans, LA, Jan 9-12), 276-279.

YANKELOVICH, N. 1996. How do users know what to say? ACM Interactions, 3, 6 (December 1996).

YANKELOVICH, N. and LAI, J. 1998. Designing speech user interfaces. In Proceedings of CHI'98 (Los Angeles, CA, April 18-23).

YANKELOVICH, N.; LEVOW, G.-A. and MARX M. 1995. Designing SpeechActs: Issues in speech user interfaces. In Proceedings of CHI'95 (Denver, CO, May 7-11).

YI, J.; GLASS, J. and HETHERINGTON, L. 2000. A Flexible, Scalable Finite-State Transducer Architecture for Corpus-Based Concatenative Speech Synthesis. In Proceedings of ICSLP.

ZADROZNY, W. 1996. Natural language processing: Structure and complexity. In Proceedings of 8th International Conference on Software Engineering and Knowledge Engineering (SEKE'96, Lake Tahoe, 1996), 595-602.

ZUE, V.; SENEFF, S.; GLASS, J.; POLIFRONI, J.; PAO, C.; HAZEN, T.J.; HETHERINGTON, L. 2000. Jupiter: A Telephone-Based Conversational Interface for Weather Information. IEEE Transactions on Speech and Audio Processing, 8, 1.

Apéndice A – Documentos de definición del entorno propuesto

Y desta verdad te pudiera traer tantos ejemplos que te cansaran
Don Quijote – Capítulo XLII – Segunda Parte

La definición del entorno propuesto (ver capítulo 3) se compone de numerosos documentos con distinta funcionalidad. En el presente apéndice, se presenta parte de los documentos más representativos que se han empleado para la creación del entorno real de demostración. En alguno de estos documentos, se han omitido ciertos parámetros o atributos para hacerlos más comprensible dentro del marco de la presente tesis doctoral.

En primer lugar, se definen los DDCEs, que especifican las clases de entidad permitidas en cualquier entorno. La definición de todo entorno se compone de una clase *raíz* (actualmente vacía) de la que heredan todas las clases presentes en el mismo. Esta clase se define en el documento DDCE/root.xml:

```
<classes>
   <class name="root"/>
</classes>
```

De esta clase *raíz*, heredan diversas clases genéricas. Las que mayor interés tienen para el presente trabajo son:

- El documento DDCE/device.xml, que engloba a los distintos tipos de dispositivos que pueden estar presentes en un entorno:

```
<classes>
   <class name="device" extends="root"/>
</classes>
```

- El documento DDCE/image.xml, que define las diferentes imágenes que se pueden mostrar en las pantallas del entorno:

```
<classes>
   <class name="image" extends="root">
      <property name="Fuente"/>
   </class>
</classes>
```

- El documento DDCE/person.xml, que especifica los atributos básicos que puede tener una persona:

```
<classes>
   <class name="person" extends="root">
      <property name="Nombre"/>
      <property name="Apellidos"/>
      <property name="NumTarjeta"/>
      <property name="Esta"/>
      <property name="Correoe"/>
   </class>
</classes>
```

- Y el documento DDCE/room.xml, donde se representan las diferentes estancias de un entorno:

```
<classes>
   <class name="room" extends="root"/>
</classes>
```

A partir de la clase *image* se define la clase *picture* en el documento DDCE/picture.xml. Esta clase permite modelar cuadros dinámicos presentes en el entorno:

```
<classes>
   <class name="picture" extends="image">
      <property name="Nombre_Cuadro"/>
      <property name="Autor"/>
      <property name="Año_Pintado"/>
      <property name="Nacionalidad"/>
   </class>
</classes>
```

Uno de los elementos principales del entorno son los dispositivos. Algunos de estos dispositivos, que heredan de la clase *device*, se definen en los siguientes documentos:

- El documento DDCE/audio_source.xml, que definen posibles fuentes de audio, como un reproductor de discos compactos o un sintonizador de radio:

```
<classes>
   <class name="audio_source" extends="device">
      <property name="Fuente"/>
      <property name="Puerto"/>
      <property name="IP"/>
      <property name="Volumen_CD"/>
      <property name="Status_CD"/>
      <property name="Escuchar_track_unico"/>
      <property name="Escuchar_track_aleatorio"/>
      <property name="Escuchar_a_partir_de_track"/>
      <property name="Escucha_continua"/>
      <property name="Pausa"/>
      <property name="Continuar"/>
      <property name="Volumen_Radio"/>
      <property name="Canal"/>
   </class>
</classes>
```

- El documento DDCE/door.xml, que define los estados abierto y cerrado y de una puerta:

```
<classes>
   <class name="door" extends="device">
      <property name="Status"/>
   </class>
</classes>
```

- El documento DDCE/speaker.xml donde se representas las características que puede presentar un altavoz:

```
<classes>
   <class name="speaker" extends="device">
      <property name="Habilitar"/>
      <property name="Ready"/>
      <property name="Status"/>
      <property name="Puerto"/>
      <property name="IP"/>
      <property name="Velocidad"/>
      <property name="Left_Volume"/>
      <property name="Right_Volume"/>
      <property name="Graves"/>
      <property name="Agudos"/>
      <property name="Master_Volume"/>
   </class>
</classes>
```

- Y el documento DDCE/light.xml donde se definen la propiedad de encendida y apagada para una luz:

```
<classes>
   <class name="light" extends="device">
      <property name="Status"/>
   </class>
</classes>
```

A su vez, de la clase *light* heredan tres nuevas clases, que especifican nuevos tipos de luces:

- El documento DDCE/dimmerlight.xml que define aquellas luces en donde además se puede regular su intensidad:

```
<classes>
   <class name="dimmerlight" extends="light">
      <property name="Value"/>
   </class>
</classes>
```

- El documento DDCE/readinglight.xml, , para focos o luces de lectura:

```
<classes>
   <class name="readinglight" extends="light"/>
</classes>
```

- Y el documento DDCE/fluorescentlight.xml, , para luces fluorescentes:

```
<classes>
   <class name="fluorescentlight" extends="light"/>
</classes>
```

Todas estas clases, se pueden parametrizar para añadir nueva información a cada una de ellas. Para esto se utilizan los DPCEs. Los casos más representativos son la parametrización relativa a la información para la creación de la interfaz Web (*Jeoffrey*) y la parametrización relacionada con la interfaz de diálogos orales (*Odisea*).

En el caso de la interfaz Web, algunos de los DPCEs empleados son:

- El documento DPCE/Jeoffrey/audio_source_xml:

```
<classes>
   <class name="audio_source">
      <property name="Fuente">
         <paramSet name="jeoffrey">
            <param name="type">choice</param>
            <param name="entry">Off NADA</param>
            <param name="entry">Lector_cd LECTOR_CD</param>
            <param name="entry">Radio RADIO</param>
            <param name="dependences">1</param>
         </paramSet>
      </property>
      <property name="Puerto">
         <paramSet name="jeoffrey">
            <param name="type">entry</param>
            <param name="enable">1.NADA</param>
         </paramSet>
      </property>
      <property name="IP">
         <paramSet name="jeoffrey">
            <param name="type">entry</param>
            <param name="enable">1.NADA</param>
         </paramSet>
      </property>
      <property name="Volumen_CD">
         <paramSet name="jeoffrey">
            <param name="type">slider</param>
            <param name="lo">0</param>
            <param name="hi">100</param>
            <param name="unit">5</param>
            <param name="enable">1.LECTOR_CD</param>
```

```
      </paramSet>
   </property>
   <property name="Status_CD">
      <paramSet name="jeoffrey">
         <param name="type">alarm</param>
         <param name="entry">-1 0xFFFFFF</param>
         <param name="entry">0 0xFF0000</param>
         <param name="entry">1 0x00FF00</param>
         <param name="entry">2 0x000000</param>
         <param name="entry">3 0x0000FF</param>
         <param name="enable">1.LECTOR_CD</param>
      </paramSet>
   </property>
   <property name="Escuchar_track_unico">
      <paramSet name="jeoffrey">
         <param name="type">entry</param>
         <param name="enable">1.LECTOR_CD</param>
      </paramSet>
   </property>
   <property name="Escuchar_track_aleatorio">
      <paramSet name="jeoffrey">
         <param name="type">entry</param>
         <param name="enable">1.LECTOR_CD</param>
      </paramSet>
   </property>
   <property name="Escuchar_a_partir_de_track">
      <paramSet name="jeoffrey">
         <param name="type">entry</param>
         <param name="enable">1.LECTOR_CD</param>
      </paramSet>
   </property>
   <property name="Escucha_continua">
      <paramSet name="jeoffrey">
         <param name="type">entry</param>
         <param name="enable">1.LECTOR_CD</param>
      </paramSet>
   </property>
   <property name="Pausa">
      <paramSet name="jeoffrey">
         <param name="type">button</param>
         <param name="text">Pausa</param>
         <param name="cmd">1</param>
         <param name="enable">1.LECTOR_CD</param>
      </paramSet>
   </property>
   <property name="Continuar">
      <paramSet name="jeoffrey">
         <param name="type">button</param>
         <param name="text">Continuar</param>
         <param name="cmd">1</param>
         <param name="enable">1.LECTOR_CD</param>
      </paramSet>
   </property>
```

```
      <property name="Volumen_Radio">
         <paramSet name="jeoffrey">
            <param name="type">slider</param>
            <param name="lo">0</param>
            <param name="hi">100</param>
            <param name="unit">5</param>
            <param name="enable">1.RADIO</param>
         </paramSet>
      </property>
      <property name="Canal">
         <paramSet name="jeoffrey">
            <param name="type">choice</param>
            <param name="entry">M80 0</param>
            <param name="entry">Alcobendas 1</param>
            <param name="entry">Radio5 2</param>
            <param name="entry">Los_40 3</param>
            <param name="entry">Radio3 4</param>
            <param name="entry">Top40 5</param>
            <param name="entry">Onda0 6</param>
            <param name="entry">Cadena100 7</param>
            <param name="entry">Cope 8</param>
            <param name="entry">Onda_Madrid 9</param>
            <param name="entry">Kiss_FM 10</param>
            <param name="entry">Maxima 11</param>
            <param name="entry">Radio1 12</param>
            <param name="entry">SER 13</param>
            <param name="enable">1.RADIO</param>
         </paramSet>
      </property>
   </class>
</classes>
```

- El documento DPCE/Jeoffrey/door.xml:

```
<classes>
   <class name="door">
      <property name="Status">
         <paramSet name="jeoffrey">
            <param name="type">button</param>
            <param name="text">Abrir puerta</param>
            <param name="cmd">1</param>
            <param name="timeout">10000</param>
            <param name="color">0x00FF00</param>
         </paramSet>
      </property>
   </class>
</classes>
```

- El documento DPCE/Jeoffrey/light.xml:

```
<classes>
   <class name="light">
      <property name="Status">
         <paramSet name="jeoffrey">
            <param name="type">switch</param>
            <param name="text_off">Encender luz</param>
            <param name="cmd_off">1</param>
            <param name="text_on">Apagar luz</param>
            <param name="cmd_on">0</param>
            <param name="color_on">0x00FF00</param>
         </paramSet>
      </property>
   </class>
</classes>
```

- El documento DPCE/Jeoffrey/dimmerlight:

```
<classes>
   <class name="dimmerlight">
      <property name="Value">
         <paramSet name="jeoffrey">
            <param name="type">slider</param>
            <param name="lo">0</param>
            <param name="hi">64</param>
            <param name="unit">4</param>
         </paramSet>
      </property>
   </class>
</classes>
```

Para la interfaz Web, los DPCEs utilizados son:

- El documento DPCE/Odisea/audio_source_xml:

```
<classes>
   <class name="audio_source">
      <property name="Fuente">
         <paramSet name="dialogue">
            <param name="action1">apagar</param>
            <param name="skeleton1">VP apagar quitar
            OP NCFS00A:radio</param>
         </paramSet>
      </property>
      <property name="Volumen_Radio">
         <paramSet name="dialogue">
            <param name="action1">subir</param>
            <param name="skeleton1">subir aumentar OP volumen
            LP NCFS00A:radio</param>
            <param name="skeleton2">VP subir aumentar
```

```
      OP NCFS00A:radio</param>
      <param name="action2">bajar</param>
      <param name="skeleton3">VP bajar disminuir OP volumen
      LP NCFS00A:radio</param>
      <param name="skeleton4">VP bajar disminuir
      OP NCFS00A:radio</param>
   </paramSet>
</property>
<property name="Canal">
   <paramSet name="dialogue">
      <param name="action1">encender_m_80</param>
      <param name="skeleton1">VP encender poner cambiar
      OP NCFS00A:radio emisora SUB MP NPCN00A:m_80</param>
      <param name="skeleton2">VP poner OP m_80</param>
      <param name="skeleton3">VP cambiar IOP m_80</param>
      <param name="action2">encender_radio_alcobendas</param>
      <param name="skeleton4">VP encender poner cambiar
      OP NCFS00A:radio emisora
      SUB MP NPCN00A:radio_alcobendas</param>
      <param name="skeleton5">VP poner
      OP radio_alcobendas</param>
      <param name="skeleton6">VP cambiar
      IOP radio_alcobendas</param>
      <param name="action3">encender_radio_5</param>
      <param name="skeleton7">VP encender poner cambiar
      OP NCFS00A:radio emisora SUB MP NPCN00A:radio_5</param>
      <param name="skeleton8">VP poner OP radio_5</param>
      <param name="skeleton9">VP cambiar IOP radio_5</param>
      <param name="action4">encender_cuarenta</param>
      <param name="skeleton10">VP encender poner cambiar
      OP NCFS00A:radio emisora SUB MP NPMP00A:cuarenta
      NPMP00A:cuarenta_principales</param>
      <param name="skeleton11">VP poner OP cuarenta
      NPMP00A:cuarenta_principales</param>
      <param name="skeleton12">VP cambiar IOP cuarenta
      NPMP00A:cuarenta_principales</param>
      <param name="action5">encender_radio_3</param>
      <param name="skeleton13">VP encender poner cambiar
      OP NCFS00A:radio emisora SUB MP NPCN00A:radio_3</param>
      <param name="skeleton14">VP poner OP radio_3</param>
      <param name="skeleton15">VP cambiar IOP radio_3</param>
      <param name="action6">encender_top_40</param>
      <param name="skeleton16">VP encender poner cambiar
      OP NCFS00A:radio emisora SUB MP NPCN00A:top_40</param>
      <param name="skeleton17">VP poner OP top_40</param>
      <param name="skeleton18">VP cambiar IOP top_40</param>
      <param name="action7">encender_onda_0</param>
      <param name="skeleton19">VP encender poner cambiar
      OP NCFS00A:radio emisora SUB MP NPCN00A:onda_0</param>
      <param name="skeleton20">VP poner OP onda_0</param>
      <param name="skeleton21">VP cambiar IOP onda_0</param>
      <param name="action8">encender_cadena_100</param>
      <param name="skeleton22">VP encender poner cambiar
```

```
            OP NCFS00A:radio emisora SUB MP NPCN00A:cadena_100</param>
            <param name="skeleton23">VP poner OP cadena_100</param>
            <param name="skeleton24">VP poner IOP cadena_100</param>
            <param name="action9">encender_cope</param>
            <param name="skeleton25">VP encender poner cambiar
            OP NCFS00A:radio emisora SUB MP NPFS00A:cope</param>
            <param name="skeleton26">VP poner OP cope</param>
            <param name="skeleton27">VP cambiar IOP cope</param>
            <param name="action10">encender_onda_madrid</param>
            <param name="skeleton28">VP encender poner cambiar
            OP NCFS00A:radio emisora SUB MP NPCN00A:onda_madrid</param>
            <param name="skeleton29">VP poner OP onda_madrid</param>
            <param name="skeleton30">VP cambiar IOP onda_madrid</param>
            <param name="action11">encender_equis_f_m</param>
            <param name="skeleton31">VP encender poner cambiar
            OP NCFS00A:radio emisora SUB MP NPCN00A:equis_f_m
            NPCN00A:equis</param>
            <param name="skeleton32">VP poner
            OP equis_f_m equis</param>
            <param name="skeleton33">VP cambiar
            IOP equis_f_m equis</param>
            <param name="action12">encender_máxima_f_m</param>
            <param name="skeleton34">VP encender poner cambiar
            OP NCFS00A:radio emisora SUB MP NPCN00A:máxima_f_m</param>
            <param name="skeleton35">VP poner OP máxima_f_m</param>
            <param name="skeleton36">VP cambiar IOP máxima_f_m</param>
            <param name="action13">encender_radio_1</param>
            <param name="skeleton37">VP encender poner cambiar
            OP NCFS00A:radio emisora
            SUB MP NPCN00A:radio_1 radio_nacional</param>
            <param name="skeleton38">VP poner
            OP radio_1 radio_nacional</param>
            <param name="skeleton39">VP cambiar
            IOP radio_1 radio_nacional</param>
            <param name="action14">encender_ser</param>
            <param name="skeleton40">VP encender poner cambiar
            OP NCFS00A:radio emisora SUB MP NPFS00A:ser</param>
            <param name="skeleton41">VP poner OP ser</param>
            <param name="skeleton42">VP cambiar IOP ser</param>
        </paramSet>
      </property>
   </class>
</classes>
```

- El documento DPCE/Odisea/door.xml:

```
<classes>
   <class name="door">
      <property name="Status">
         <paramSet name="dialogue">
            <param name="action1">abrir</param>
            <param name="skeleton1">VP abrir OP puerta</param>
         </paramSet>
      </property>
   </class>
</classes>
```

- El documento DPCE/Odisea/light.xml:

```
<classes>
   <class name="light">
      <property name="Status">
         <paramSet name="dialogue">
            <param name="action1">encender</param>
            <param name="skeleton1">VP encender OP luz</param>
            <param name="action2">apagar</param>
            <param name="skeleton2">VP apagar OP luz</param>
         </paramSet>
      </property>
   </class>
</classes>
```

- El documento DPCE/Odisea/fluorescentlight.xml:

```
<classes>
   <class name="fluorescentlight">
      <property name="Status">
         <paramSet name="dialogue">
            <param name="action1">encender</param>
            <param name="skeleton1">VP encender poner OP luz
            LP techo arriba</param>
            <param name="skeleton2">VP encender poner OP luz
            MP AQ0MS00:principal AQ0MS00:general</param>
            <param name="skeleton3">VP encender poner
            OP fluorescente</param>
            <param name="action2">apagar</param>
            <param name="skeleton4">VP apagar quitar OP luz
            LP techo arriba</param>
            <param name="skeleton5">VP apagar quitar OP luz
            MP AQ0MS00:principal AQ0MS00:general</param>
            <param name="skeleton6">VP apagar quitar
            OP fluorescente</param>
         </paramSet>
      </property>
   </class>
</classes>
```

- El documento DPCE/Odisea/readinglight.xml:

```
<classes>
   <class name="readinglight">
      <property name="Status">
         <paramSet name="dialogue">
            <param name="action1">encender</param>
            <param name="skeleton1">VP encender poner
            OP luz NCFS000:lámpara MP lectura</param>
            <param name="skeleton2">VP encender poner OP luz
            LP foco</param>
            <param name="skeleton3">VP encender poner OP foco</param>
            <param name="action2">apagar</param>
            <param name="skeleton4">VP apagar quitar OP luz
            NCFS000:lámpara MP lectura</param>
            <param name="skeleton5">VP apagar quitar OP luz
            LP foco</param>
            <param name="skeleton6">VP apagar quitar OP foco</param>
         </paramSet>.
      </property>
   </class>
</classes>
```

- El documento DPCE/Odisea/dimmerlight.xml:

```
<classes>
   <class name="dimmerlight">
      <property name="Status">
         <paramSet name="dialogue">
            <param name="action1">encender</param>
            <param name="skeleton1">VP encender poner
            OP luz NCFS000:lámpara MP ambiente halógena</param>
            <param name="action2">apagar</param>
            <param name="skeleton2">VP apagar quitar
            OP luz NCFS000:lámpara MP ambiente halógena</param>
         </paramSet>
      <property name="Value">
         <paramSet name="dialogue">
            <param name="action1">subir</param>
            <param name="skeleton1">VP subir aumentar
            OP luz NCFS000:lámpara MP ambiente halógena</param>
            <param name="skeleton2">VP subir aumentar OP intensidad
            LP luz NCFS000:lámpara MP ambiente halógena</param>
            <param name="action2">bajar</param>
            <param name="skeleton3">VP bajar disminuir reducir
            OP luz NCFS000:lámpara MP ambiente halógena</param>
            <param name="skeleton4">VP bajar disminuir reducir
            OP intensidad LP luz NCFS000:lámpara
            MP ambiente halógena</param>
         </paramSet>
      </property>
   </class>
</classes>
```

Una vez definidos los elementos comunes a todos los entornos se empieza a definir un entorno concreto y sus elementos utilizando los DDEEs. Entre ellos, se puede resaltar:

- El documento DDEE/devices.xml donde se especifican todos los dispositivos hallados en las diferentes estancias del entorno:

```
<instances>
   <!-- Entidad Luz del salón -->
   <entity name="Lamp_1" type="fluorescentlight"/>
   <!-- Entidad Lampara de pie 1 -->
   <entity name="LampV1D" type="dimmerlight"/>
   <!-- Entidad Lampara de pie 2 -->
   <entity name="LampV2D" type="dimmerlight"/>
   <!-- Entidad Lampara de lectura 1 -->
   <entity name="LampV1R" type="readinglight"/>
   <!-- Entidad Lampara de lectura 2 -->
   <entity name="LampV2R" type="readinglight"/>
   <!-- Entidad Puerta -->
   <entity name="Puerta1" type="door"/>
   <!-- Entidad Detector de la puerta -->
   <entity name="Detector_Puerta" type="sensor"/>
   <!-- Entidad Lector de tarjeta -->
   <entity name="Tarjeta" type="card_reader"/>
   <!-- Entidad Altavoz 1 -->
   <entity name="Altavoz" type="speaker"/>
   <!-- Entidad Altavoz 3 -->
   <entity name="Altavoz3" type="speaker"/>
   <!-- Entidad generador de música -->
   <entity name="Genmusic" type="audio_source"/>
   <!-- Entidad WebCam -->
   <entity name="webCam" type="webcam"/>
   <!-- Face Recognizer -->
   <entity name="frecog" type="frecognizer"/>
</instances>
```

- El documento DDEE/pictures.xml que establece los cuadros que se pueden mostrar dentro del entorno:

```
<instances>
   <!-- Entidad El nacimiento de Venus -->
   <entity name="nacimiento_Venus" type="picture"/>
   <!-- Entidad La Persistencia de la Memoria -->
   <entity name="Persistencia_Memoria" type="picture"/>
   <!-- Entidad Mona Lisa -->
   <entity name="Mona_Lisa" type="picture"/>
   <!-- Entidad El 2 de Mayo -->
   <entity name="El_2_Mayo" type="picture"/>
   <!-- Entidad El 3 de Mayo -->
   <entity name="El_3_Mayo" type="picture"/>
   <!-- Entidad La familia de Carlos IV -->
   <entity name="familia_Carlos_IV" type="picture"/>
```

```
    <!-- Entidad La Maja Vestida -->
    <entity name="Maja_Vestida" type="picture"/>
    <!-- Entidad Composición I -->
    <entity name="Composicion_I" type="picture"/>
    <!-- Entidad El puente de Argenteuil -->
    <entity name="Puente_Argenteuil" type="picture"/>
    <!-- Entidad  La Regata de Argenteuil -->
    <entity name="Regata_Argenteuil" type="picture"/>
    <!-- Entidad Las Barcas -->
    <entity name="Barcas" type="picture"/>
    <!-- Entidad Mujeres en el jardín -->
    <entity name="Mujeres_jardin" type="picture"/>
    <!-- Entidad Las amapolas -->
    <entity name="amapolas" type="picture"/>
    <!-- Entidad El grito -->
    <entity name="grito" type="picture"/>
    <!-- Entidad La Concepción del Escorial -->
    <entity name="Concepcion" type="picture"/>
    <!-- Entidad Paisaje -->
    <entity name="Paisaje" type="picture"/>
    <!-- Entidad Guernica -->
    <entity name="guernica" type="picture"/>
    <!-- Entidad La Familia de saltimbanquis -->
    <entity name="saltimbanquis" type="picture"/>
    <!-- Entidad En el jardín -->
    <entity name="jardin" type="picture"/>
    <!-- Entidad meninas -->
    <entity name="meninas" type="picture"/>
    <!-- Entidad Las lanzas -->
    <entity name="lanzas" type="picture"/>
    <!-- Entidad A las afueras de París -->
    <entity name="afueras_Paris" type="picture"/>
    <!-- Entidad El Sena con el Puente de la Grande Jatte -->
    <entity name="Sena_Puente_Grande_Jatte" type="picture"/>
    <!-- Entidad Print gallery -->
    <entity name="Print_gallery" type="picture"/>
    <!-- Entidad Waterfall -->
    <entity name="Waterfall" type="picture"/>
    <!-- Entidad Other world II -->
    <entity name="Other_world_II" type="picture"/>
    <!-- Entidad Drawing hands -->
    <entity name="Drawing_hands" type="picture"/>
    <!-- Entidad  Doric Columns-->
    <entity name="Doric_Columns" type="picture"/>
    <!-- Entidad  Ascending and descending -->
    <entity name="Ascending_descending" type="picture"/>
</instances>
```

- El documento DDEE/persons.xml, donde se determinan las personas que tienen acceso al entorno junto con los cuadros que prefieren visualizar cuando están presentes en el mismo:

```
<instances>
   <!-- Entidad Xavier Alamán -->
   <entity name="xavier" type="person">
      <relation name="likes" destination="meninas"/>
      <relation name="likes" destination="guernica"/>
      <relation name="likes" destination="Ascending_descending"/>
   </entity>
   <!-- Entidad Germán Montoro -->
   <entity name="german" type="person">
      <relation name="likes" destination="Maja_Vestida"/>
      <relation name="likes" destination="Puente_Argenteuil"/>
      <relation name="likes" destination="Concepcion"/>
   </entity>
   <!-- Entidad Pablo Haya -->
   <entity name="phaya" type="person">
      <relation name="likes" destination="nacimiento_Venus"/>
      <relation name="likes" destination=" Waterfall"/>
   </entity>
</instances>
```

- Y el documento DDEE/rooms.xml donde se especifican las habitaciones que componen el entorno y los elementos presentes en cada una de ellas:

```
<instances>
   <entity name="lab_B403" type="room">
   <!-- Dispositivos dentro de la habitación B403 -->
   <relation name="has_resource" destination="Lamp_1"/>
   <relation name="has_resource" destination="Switch0_1"/>
   <relation name="has_resource" destination="Switch1_1"/>
   <relation name="has_resource" destination="SwitchX_1"/>
   <relation name="has_resource" destination="SwitchX_2"/>
   <relation name="has_resource" destination="Led _1"/>
   <relation name="has_resource" destination="LampV1D"/>
   <relation name="has_resource" destination="LampV2D"/>
   <relation name="has_resource" destination="LampV1R"/>
   <relation name="has_resource" destination="LampV2R"/>
   <relation name="has_resource" destination="Puerta1"/>
   <relation name="has_resource" destination="Detector_Puerta"/>
   <relation name="has_resource" destination="Altavoz1"/>
   <relation name="has_resource" destination="Altavoz3"/>
   <relation name="has_resource" destination="Genmusic"/>
   <relation name="has_resource" destination="Tarjeta"/>
   </entity>
<instances>
```

Por último, se pueden parametrizar las propiedades de estos elementos para adaptarlas al entorno propuesto. Para ello se emplean los DPEEs. Como ocurría con los DPCEs, este proceso se puede realizar para la interfaz Web y la de diálogos orales.

En el caso de la interfaz Web, se emplea un documento por dispositivo para especificar la imagen que se utiliza para representarlo dentro de la misma y sus coordenadas. Así ocurre, por ejemplo, con el documento DPEE/Jeoffrey/genmusic.xml:

```
<instances>
   <entity name="Genmusic">
      <paramSet name="jeoffrey">
         <param name="image">cadena.gif</param>
         <param name="x">395</param>
         <param name="y">315</param>
      </paramSet>
   </entity>
</instances>
```

Además, se especifica el fondo que sirve de plantilla para representar el entorno en la interfaz y su tamaño en el documento DPEE/Jeoffrey/lab_B403.xml:

```
<instances>
   <entity name="lab_B403">
      <paramSet name="jeoffrey">
         <param name="width">764</param>
         <param name="height">513</param>
         <param name="background">fondof.jpg</param>
      </paramSet>
   </entity>
</instances>
```

Para la interfaz de diálogos orales, se parametrizan aquellas entidades que presenten alguna información lingüística adicional o diferente en el entorno propuesto. Así ocurre con:

- El documento DPEE/Odisea/lampv1r.xml:

```
<instances>
   <entity name="lampV1R">
      <property name="status">
         <paramSet name="dialogue">
            <param name="add_all">LP izquierda</param>
         </paramSet>
      </property>
   </entity>
</instances>
```

- El documento DPEE/Odisea/lampv2r.xml:

```
<instances>
   <entity name="lampV2R">
      <property name="status">
         <paramSet name="dialogue">
            <param name="add_all">LP derecha</param>
         </paramSet>
      </property>
   </entity>
</instances>
```

- El documento DPEE/Odisea/lampv1d.xml:

```
<instances>
   <entity name="lampV1D">
      <property name="status">
         <paramSet name="dialogue">
            <param name="add_all">LP izquierda</param>
         </paramSet>
      </property>
   </entity>
</instances>
```

- Y el documento DPEE/Odisea/lampv2d.xml:

```
<instances>
   <entity name="lampV2D">
      <property name="status">
         <paramSet name="dialogue">
            <param name="add_all">LP derecha</param>
         </paramSet>
      </property>
   </entity>
</instances>
```

Todos estos documentos se unen para formar un único Documento de Definición del Entorno, que establece las características concretas del entorno propuesto y permite crear de forma automática la capa *middleware* y las interfaces de interacción con el mismo.

Apéndice B – Métodos BBAction y hasRequestedADifferentAction

Las acciones que ni mudan, ni alteran la verdad de la historia, no hay para qué escribirlas
Don Quijote – Capítulo III – Segunda Parte

En el presenta apéndice se muestra un ejemplo de los métodos *BBAction* y *hasRequestedADifferentAction* para una entidad de tipo *dimmerlight* (una luz regulable) en el entorno propuesto. La implementación de estas funciones se ha realizado utilizando el lenguaje de programación *Java*.

Ambos métodos reciben la oración pronunciada por el usuario (*sentence*) y la acción o acciones requeridas (*actionNames*). A su vez, se valen de funciones genéricas proporcionadas que facilitan su implementación. Algunas de estas son:

- *getStatus*: que consulta la pizarra y devuelve el valor del estado actual de la entidad en el entorno.
- *setStatus*: que escribe en la propiedad valor de la entidad de la pizarra el dato que recibe como argumento. De este modo modifica el estado físico del elemento en el entorno.
- *synthesis.say*: que sintetiza y pronuncia la oración recibida como argumento.

El código de la función *BBAction* implementada para las entidades de tipo *dimmerlight* es:

```
public int BBAction(TreeSentence sentence, Vector actionNames)
{
   /* Se obtiene la acción solicitada por el usuario */
   String actionName = (String)actionNames.elementAt(0);

   /* Se obtiene el estado actual de la luz */
   int value = getStatus();

   /* La luz está al máximo y el usuario solicita encenderla
   o subirla: se informa al usuario */
   if ( (value == 50) && ( actionName.equalsIgnoreCase("subir")
   || actionName.equalsIgnoreCase("encender") ) )
   {
      synthesis.say("La luz ya está al máximo");
      return OK;
   }
   /* La luz ya está encendida y el usuario solicita encenderla:
   se aumenta su intensidad */
   else if ((value > 0) && ( actionName.equalsIgnoreCase("encender")))
   {
      value += 10;
      if (value > 50)
      value = 50;
      /* Se envía la información a la pizarra */
      setStatus(new Integer(value).toString());
   }
   /* La luz está apagada y el usuario solicita apagarla:
   se le informa */
   else if ( (value == 0) && (actionName.equalsIgnoreCase("apagar")) )
   {
      synthesis.say("La luz ya está apagada");
```

```
      return OK;
   }
   /* La luz está apagada y el usuario solicita bajarla:
   se le informa */
   else if ( (value == 0) && (actionName.equalsIgnoreCase("bajar")) )
   {
      synthesis.say("La luz ya está al mínimo");
      return OK;
   }
   /* La luz está apagada y el usuario solicita encenderla:
   se enciende */
   if ( actionName.equalsIgnoreCase("encender") )
   {
      value = 20;
      setStatus(new Integer(value).toString());
   }
   /* La luz está encendida y el usuario solicita apagarla:
   se apaga */
   else if ( actionName.equalsIgnoreCase("apagar") )
   {
      value = 0;
      setStatus(new Integer(value).toString());
   }
   /* El usuario solicita subir la luz: se sube */
   else if ( actionName.equalsIgnoreCase("subir") )
   {
      value += 10;
      if (value > 50)
      value = 50;
      setStatus(new Integer(value).toString());
   }
   /* La luz está encendida y el usuario solicita bajarla: se baja */
   else if ( actionName.equalsIgnoreCase("bajar") )
   {
      value -= 10;
      if (value < 0)
      value = 0;
      setStatus(new Integer(value).toString());
   }
   return OK;
}
```

El código de la función *hasRequestedADifferentAction* implementada para las entidades de tipo *dimmerlight* es:

```
public boolean hasRequestedADifferentAction(TreeSentence sentence,
Vector actionNames)
{
   /* Se obtiene la acción solicitada por el usuario */
   String actionName = (String)actionNames.elementAt(0);

   /* Casos donde la acción solicitada no es diferente
   al estado actual: */

   /* Si la acción es bajar, apagar o subir y la luz está apagada */
   if ( ( actionName.equalsIgnoreCase("bajar") ||
   actionName.equalsIgnoreCase("apagar") ||
   actionName.equalsIgnoreCase("subir") ) && (getStatus()==0) )
      return false;

   /* Si la acción es encender y la luz está encendida */
   if ( actionName.equalsIgnoreCase("encender") && (getStatus()>0) )
      return false;

   /*En los demás casos, se ha seleccionado una acción diferente*/
   return true;
}
```

Apéndice C – Índice de acrónimos

"Sé breve en tus razonamientos, que ninguno hay gustoso si es largo"
Don Quijote – Capítulo XXI – Primera Parte

Aire - Agent-Based Intelligent Reactive Environments. Página 24

API - Application Programming Interface. Páginas 32, 69, 90, 91, 156, 156, 157

CSLU - Center for Spoken Language Understanding. Página 101

DDCE – Documento de Descripción de las Clases de Entidad. Páginas 70, 71, 72, 74, 76, 83, 157.

DDE – Documento de Descripción del Entorno. Páginas 76, 77, 77, 82, 110, 117, 120, 158.

DDEE – Documento de Descripción de las Entidades del Entorno. Páginas 72, 73, 74, 76, 110, 110, 115, 116, 157.

DFKI - Centro Alemán de Investigación de Inteligencia Artificial. Páginas 16, 62

DL – Lenguaje de descripción. Páginas 13, 69

DPCE – Documento de Particularización de las Clases de Entidad. Páginas 71, 72, 73, 74, 76, 83, 110, 111, 113, 116, 117, 125, 157.

DPEE – Documento de Particularización de las Entidades del Entorno. Páginas 73, 74, 75, 84, 115, 116, 117, 157, 157.

DTD - Document Type Declaration. Página 64

DTMF – Dual Tone Multi-Frequency. Páginas 100, 100

EIB - European Installation Bus. Páginas 68, 68, 81, 180

GUI – Interfaz gráfica de usuario. Página 49

HTTP - HyperText Transfer Protocol. Páginas 69, 78, 156, 156

INRIA - Institut National de Recherche en Informatique et en Automatique. Páginas 16, 33

IP – Internet Protocol. Página 80

JSGF – Java Speech Grammar Format. Páginas 91, 118

KQML - Knowledge Query and Manipulation Language. Página 64

LGPL - Lesser General Public License. Página 101

MAP – Módulo de Adición de Palabras. Páginas 128, 129, 130, 150, 152

MI - Mixed Initiative. Páginas 49, 50

MIO - Módulo de Intercambio de Oraciones. Páginas 97, 97

MIT - Massachusetts Institute of Technology. Páginas 16, 20, 21, 22, 28, 34, 58, 59

MMR – Módulo de Mapeo del Resultado. Páginas 128, 129, 130, 131, 132, 139, 142, 143, 144, 145, 146, 147, 148

MPA – Módulo de Procesamiento del Árbol. Páginas 128, 129, 130, 134, 140, 142, 142, 144, 145, 148, 149, 151

MPN – Módulo de Procesamiento de Nodos. Páginas 128, 129, 130, 131, 132, 134, 139, 140, 142, 143, 155

MPR – Módulo de Procesamiento del Resultado. Páginas 128, 129, 130, 131, 132, 133, 139, 139, 141, 143, 144, 147, 150, 152

MSD – Módulo de Selección del Diálogo. Páginas 97, 98

NAP – Número de Acciones Posibles. Páginas 130, 131

NTA – Número Total de Acciones. Páginas 130, 131, 132

OCF - Oración de Clarificación Final. Páginas 130, 131, 132

OCG – Oración de Clarificación Generada. Páginas 130, 131

PDA - Personal Digital Assistant. Páginas 32, 32, 36, 186

QUD - Questions Under Discussion. Página 55

RAD – Desarrollo Rápido de Aplicaciones. Página 101

RFID - Radio Frequency Identification. Página 20

SI - System Initiative. Páginas 49, 50

SMNP - Simple Management Network Protocol. Página 69

SQL - Structured Query Language. Página 58

SUI – Interfaz de usuario oral. Páginas 44, 44, 45, 49

Sundial - Speech UNderstanding in DIALogue. Página 56

TTS – Text To Speech. Página 89

URL - Uniform Resource Locator. Página 36

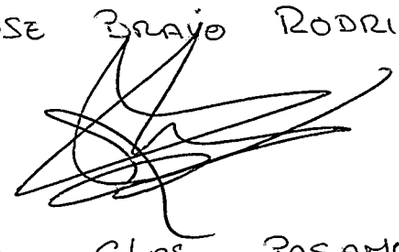
VoiceXML - Voice eXtensible Markup Language. Páginas 100, 100, 102

XML - eXtensible Markup Language. Páginas 30, 63, 64, 68, 69, 70, 76, 77, 78, 78, 83, 86, 99, 102, 156, 156, 181

Reunido el tribunal que suscribe en el día de la fecha, acordó calificar la presente Tesis doctoral con SOBRESALIENTE CUM LAUDE

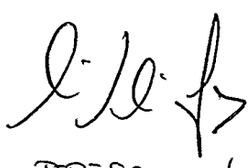
Madrid, 6-MAYO-2005

FDO: JOSE BRAVO RODRIGUEZ

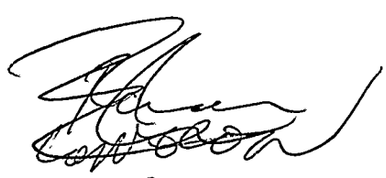


FDO: JOSE GOLAS PASAMONTES

FDO: RAMON LOPEZ-GOZAR DELGADO



FDO: MARIA TERESA LOPEZ SOTO



FDO: FCO. JAVIER GOMEZ ARRIBAS