

## ARQUITECTURA DISTRIBUIDA PARA EL DESARROLLO DE SISTEMAS DE DIÁLOGO HABLADO, EDECÁN.

*José Enrique García, Alfonso Ortega, Antonio Miguel y Eduardo Lleida.*

Grupo de Tecnologías de las Comunicaciones (GTC)  
I3A, Universidad de Zaragoza  
{jegarlai,ortega,amiguel,lleida}@unizar.es

### RESUMEN

El proyecto EDECÁN (TIN2005-08660-C04, [www.EDECAN.es](http://www.EDECAN.es)) tiene como objetivo aumentar la robustez de un sistema de diálogo de habla espontánea a través del desarrollo de tecnologías para la adaptación y personalización del mismo a los distintos contextos acústicos y de aplicación en los que pueda encontrarse. Con ese objetivo, se propone el uso de una arquitectura distribuida y flexible que permita la cooperación entre diferentes sistemas (módulos de comprensión, reconocedores, gestores de diálogo, etc. desarrollados para diferentes entornos de uso). EDECÁN hace uso de una interfaz de comunicaciones entre módulos capaz de soportar cualquier tipo de servicio en los sistemas operativos Windows y Linux. En este trabajo se describe la arquitectura, los elementos constitutivos de la misma, los procedimientos de gestión, los protocolos y los servicios desarrollados bajo este paradigma.

### 1. INTRODUCCIÓN

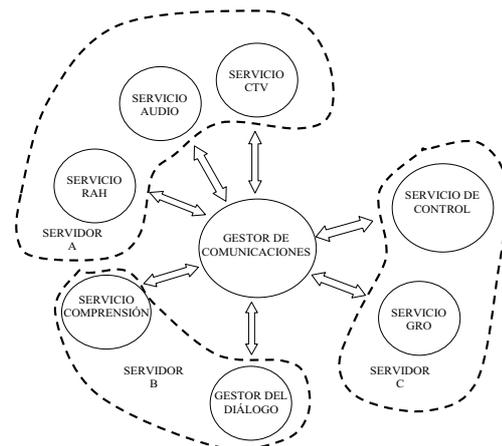
El desarrollo de aplicaciones basadas en las tecnologías del habla ha crecido enormemente en los últimos años. Junto a este crecimiento ha aumentado también la complejidad del software de las mismas, llevándose a cabo un conjunto de iniciativas para el desarrollo de plataformas y arquitecturas que facilitasen su construcción. Como ejemplos, podemos citar la arquitectura GALAXY [1] del MIT bajo la cual se han desarrollado los sistemas JUPITER (información meteorológica) [2], VOYAGER, (información urbana), u ORION (asistente personal) y que ha servido de plataforma para el proyecto DARPA COMMUNICATOR (información de viajes en avión, hotel y alquiler de coches) [3]. Basadas en los principios de GALAXY podemos encontrar otros sistemas como el CU Sonic Spine System de la Universidad de Colorado [4] o el propio sistema presentado, EDECÁN.

La filosofía de la interfaz de comunicaciones EDECÁN consiste en proveer al desarrollador de sistemas distribuidos de diálogo hablado de un conjunto de herramientas sencillas y multiplataforma, que permitan la interacción entre los distintos módulos de la

aplicación. Así, es posible el desarrollo de un nuevo módulo integrante de un sistema de diálogo hablado para enviar/recibir datos a través de escrituras/lecturas en una determinadas posiciones de memoria, sin tener que preocuparse del uso de librerías de comunicaciones propias del sistema operativo y del lenguaje de programación.

Al igual que GALAXY, la arquitectura EDECÁN presenta un nodo central configurable que permite un control flexible de las interacciones entre módulos de la aplicación de diálogo a través del establecimiento de un protocolo de comunicaciones y gestión. Además, ofrece un conjunto de librerías para el rápido desarrollo de los módulos que componen un determinado sistema.

El presente artículo está organizado del siguiente modo: En la Sección 2 se realiza una breve descripción de la arquitectura EDECÁN. La Sección 3 se dedica a la presentación del protocolo de comunicaciones, la gestión y las librerías EDECÁN. En la Sección 4 se ofrece una descripción de servicios disponibles y en la Sección 5 se apuntan las líneas futuras de evolución. Por último, en la Sección 6 se presentan las conclusiones.



**Figura 1.** Esquema de un sistema con arquitectura EDECÁN con 7 servicios repartidos en 3 servidores.

### 2. LA ARQUITECTURA EDECÁN

La arquitectura EDECÁN como se muestra en la Figura 1, está formada por un nodo central que permite la comunicación entre servidores conectados a él a través del envío de mensajes. Este nodo central actúa

Este trabajo ha sido parcialmente financiado a través del proyecto TIN2005-08660-C04.

como una batería de clientes y hace funciones de *gestor de comunicaciones*, iniciando los distintos módulos que componen el sistema y llevando a cabo el enrutado de paquetes entre ellos con dos modos de funcionamiento, a través de encaminamiento forzado, si la cabecera del paquete cuenta con un destino prefijado o mediante encaminamiento por defecto a través de la creación de una tabla de rutas configurable. El gestor de comunicaciones puede ser controlado mediante otro módulo del sistema, el cual se conecta a él como cliente y le envía comandos para realizar modificaciones sobre el sistema, como conectar servicios, desconectarlos, dar de alta nuevos servicios, o modificar la tabla de rutas.

Cada uno de los componentes del sistema (*service*) es un servicio genérico sobre el que puede instanciarse cualquier tipo de módulo (RAH, comprensión, gestor de diálogo,...). Son capaces de enviar y recibir datos y están implementados de tal manera que la máquina en la que residen (*servidor*) dispone de un proceso de fondo o demonio (*super-server*) que se ejecuta de forma continua en segundo plano y acepta las conexiones entrantes del gestor, lanzando cada uno de los servicios en un nuevo proceso y dejando la responsabilidad de la gestión de la conexión al propio servicio.



Figura 2. Ejemplo de fichero de servicios disponibles y ejemplos de configuración de algunos servicios.

Los servicios disponibles en una determinada máquina se definen a partir de un fichero de configuración en formato XML. Esta lista de servicios potenciales será consultada por el servicio genérico *service* para llevar a cabo la instanciación del servicio concreto. De este modo, cada módulo del sistema de diálogo distribuido se define a partir de una librería dinámica y un fichero de configuración en formato XML. Tanto el fichero de servicios disponibles como ejemplos de ficheros de configuración de servicios se pueden ver en la Figura 2. La librería dinámica contendrá las funciones necesarias para que el módulo en cuestión sea capaz de realizar su cometido y el fichero de configuración contiene el conjunto de

comandos a los que el servicio es capaz de responder y el nombre de la función asociada a dicha respuesta.

### 3. PROTOCOLO DE COMUNICACIONES, GESTIÓN Y LIBERÍAS EDECÁN.

EDECÁN está construida a partir de conexiones TCP/IP aunque para determinados servicios también está previsto el uso del protocolo UDP. El modo principal de funcionamiento es a través del envío de paquetes de texto en formato XML aunque también dispone de un modo de transmisión binario para el envío de información que demande un ancho de banda elevado como puede ser la voz.

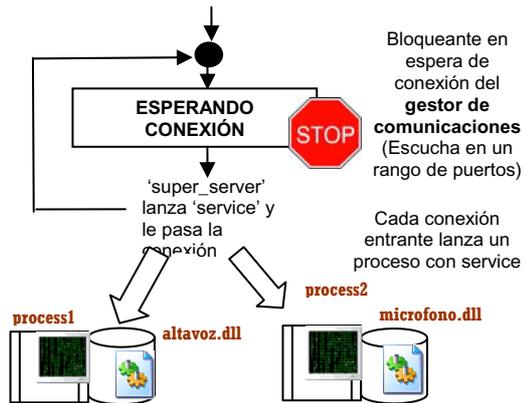


Figura 3. Diagrama de flujo del funcionamiento de un servidor EDECÁN.

#### 3.1. Protocolo de Comunicaciones.

Fundamentalmente, se establece un protocolo de comando-respuesta entre los diferentes módulos de modo que el servicio receptor ejecutará una determinada acción como respuesta a la recepción de un determinado comando. Con el objetivo de simplificar al máximo la creación de módulos, la arquitectura cuenta con el proceso *service* que gestionara las comunicaciones de cada uno de los servicios haciendo transparente el envío y recepción de paquetes desde o hacia los servicios. Cada módulo se establece como un servicio genérico que, tras la recepción del primer paquete de inicialización, se instancia como un servicio concreto a través de la carga de la librería dinámica que contiene las funciones de dicho servicio concreto. Este proceso aparece ilustrado en la Figura 3.

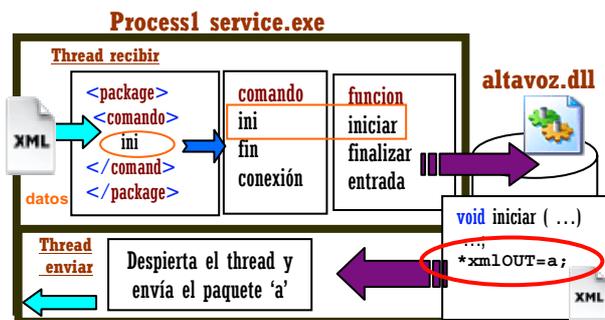


Figura 4. Proceso de recepción de comandos y ejecución de funciones en un servicio.

Una vez instanciado el servicio su modo de funcionamiento sigue el patrón mostrado en la Figura 4. En modo recepción, se comunicará con las funciones de la librería dinámica a través del paso de información mediante el uso de memoria compartida. Al recibir un paquete de otro módulo, este es interpretado y tras extraer el comando que contiene, se ejecuta su función asociada. Después se realiza el paso del resto de la información contenida en el paquete a través de estructuras de memoria compartida. En cuanto a la transmisión de paquetes, el proceso encargado de la gestión de las conexiones realiza una continua revisión de los paquetes que resultan de la ejecución de alguna de las funciones de la librería procediendo al envío de aquellos que se encuentran pendientes.

### 3.2. Biblioteca de Funciones.

El Software Development Kit (SDK) de EDECÁN cuenta también con una biblioteca de funciones que hace más simple la construcción, el envío y la recepción e interpretación de paquetes XML. En ella se encuentran definidos un conjunto de paquetes genéricos de uso muy común como pueden ser los de conexión, desconexión, etc. junto con un conjunto de funciones para crear paquetes más específicos

### 3.3. Gestión del Sistema Distribuido.

Cualquier sistema distribuido construido a partir de la arquitectura EDECÁN puede ser gestionado a través de una intuitiva y sencilla interfaz gráfica de usuario (GUI) mostrada en la Figura 5, a través de la cual pueden enviarse órdenes de configuración al gestor de comunicaciones, para que éste las ejecute. Las principales órdenes que puede dar son aquellas relacionadas con el montaje/desmontaje de un nuevo sistema, alta/baja o conexión/desconexión de un servicio, modificación la tabla de rutas del sistema o de un servicio, peticiones de información de los servicios del sistema o de las rutas del sistema. La GUI desarrollada también permite visualizar el estado de los diferentes módulos y la tasa de transmisión de subida/bajada de cada uno de ellos.

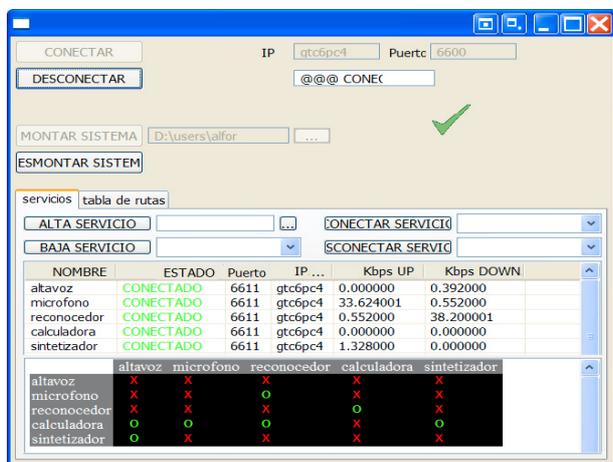


Figura 5. Interfaz Gráfica de Usuario para control y monitorización de una aplicación.



Figura 6. Ejemplo de fichero de configuración de un sistema con la arquitectura EDECÁN.

La construcción de un nuevo sistema, una vez implementados los diferentes módulos que lo componen, es tan simple como el envío de un fichero en formato XML que contenga todos los datos necesarios (IP, puerto, tipo de servicio, etc.) de cada uno de los módulos constituyentes, como se ilustra en la Figura 6. Junto a esta información, se envía la tabla de rutas a través de la cual se llevará a cabo el encaminamiento por defecto en el gestor de comunicaciones. Dicho fichero será enviado al gestor de comunicaciones encargado del montaje del sistema. A partir de ese momento, se encargará de enviar los diferentes paquetes de conexión e inicialización de los diferentes servicios a las máquinas en las que residirán estos.

## 4. LISTADO DE SERVICIOS.

Dado que el objetivo último del desarrollo del SDK EDECÁN es la investigación, desarrollo y creación de sistemas de diálogo hablado, los servicios implementados hasta el momento se han centrado principalmente en este ámbito. Sin embargo, las potencialidades del *Middleware* desarrollado trascienden este ámbito y podrán ser desarrollados módulos en cualquier otro campo de manera fácil y sencilla.

### 4.1. Servicios Básicos para Sistemas de Diálogo Hablado.

En este sentido, se han desarrollado módulos constitutivos de un sistema de diálogo hablado como un elemento capaz de capturar audio a partir del dispositivo de sonido de la máquina en la que se encuentra y transmitirlo a cualquier otra ubicación haciendo uso de un amplio abanico de modos de codificación. Por otro lado se ha implementado el servicio dual encargado de la recepción y posterior reproducción del audio. Con estos dos servicios pueden desarrollarse sencillas aplicaciones de Voz sobre IP (VoIP). Asimismo, se cuenta con sistemas de reconocimiento automático del habla, módulos de comprensión, de gestión del diálogo y de generación de respuestas orales que operan en este paradigma definido. También se ha creado un servicio

que, a partir de un motor de conversión texto-voz, envía las muestras de voz codificadas a otros módulos del sistema. Además, se cuenta con servicios que hacen posible la comunicación entre lugares remotos vía texto o módulos capaces de realizar las tareas de gestión y monitorización de un sistema ya montado.

#### 4.2. EDECÁN Mobile.

La estructura EDECÁN cuenta con un servicio especial que permite la extensión de sus usos a dispositivos portátiles (PDA, teléfonos móviles,...). Este tipo de dispositivos reciben una consideración especial debido a su limitación computacional y a la posibilidad de contar con un ancho de banda reducido. Con el objetivo de salvar estos inconvenientes, se estableció el uso de paquetes binarios mediante conmutación de paquetes en circuito virtual. Por otro lado, se desarrollaron servicios específicos como puede ser un Front-End de extracción y compresión de parámetros acústicos para arquitecturas de coma fija a partir del estándar ETSI ES 201 108.

Otro inconveniente que habitualmente presentan estos dispositivos es su conexión a través de políticas DHCP. Esto hace que la dirección IP asignada no siempre sea la misma por lo que no procede que éstos actúen como servidor. Para solucionar esto, se desarrolló un servicio puente que permite la conexión con servicios instalados en dispositivos móviles sin modificar la estructura ni la filosofía EDECÁN. De este modo, el servicio puente residente en una máquina con dirección IP estable actúa como servidor de cara al gestor de comunicaciones pero a su vez posibilita que el dispositivo móvil se conecte a él como cliente. Así, no es necesario el conocimiento previo de la dirección IP asignada al dispositivo móvil. Una vez establecidas las conexiones, el servicio puente sólo deberá reenviar los paquetes recibidos hacia o desde el cliente móvil, siendo así transparente su funcionamiento.

#### 4.3. Otros Servicios Disponibles.

Entre los servicios desarrollados para esta arquitectura, cabe destacar un módulo que detecta la presencia de una persona gracias al uso de una *web-cam* mediante un software de detección de caras, algo que puede resultar de gran utilidad para saber si alguien tiene la intención de dirigirse al sistema de diálogo hablado. También se ha desarrollado un completo *web-browser* capaz de mostrar aquellas páginas web que un usuario, a través del envío de comandos orales mediante un sistema de diálogo basado en la arquitectura EDECÁN, vaya solicitando. Asimismo, se cuenta con un módulo que realiza una adaptación de información proveniente en formato XML a una plantilla predefinida con XSLT para generar un documento HTML que podrá ser visualizado por cualquier *web-browser* con el formato deseado. Por último se han implementado módulos de adaptación de los modelos acústicos al locutor capaces de personalizar cualquiera de las aplicaciones basadas en voz que sean implementadas

bajo la arquitectura EDECÁN y módulos de normalización de vectores de características para la adaptación al entorno acústico de acuerdo con el algoritmo MEMLIN [5].

#### 5. EVOLUCIONES FUTURAS.

Como posibles evoluciones futuras se plantea la modificación de los gestores de comunicaciones para que éstos sean capaces de dar soporte a su interconexión. También se llevará a cabo la implementación de un gestor de comunicaciones más robusto, que pueda gestionar las desconexiones eventuales, de modo que sea capaz de reconectar a los servicios cuando estos se caigan.

Por último, se prevé el desarrollo de una nueva figura, un *Directorio Público de Servicios* (DPS), que aglutine los nombres, funcionalidades, direcciones IP, puertos y toda la información necesaria para implementar sistemas bajo la arquitectura EDECÁN. Así, cuando un gestor de comunicaciones solicita un determinado servicio, el DPS le proporciona la lista de todos los disponibles, junto con sus características, su carga de trabajo actual, etc., indicando igualmente el servidor correspondiente donde se hallan.

#### 6. CONCLUSIONES

En el presente trabajo se ha presentado la arquitectura desarrollada dentro del proyecto EDECÁN. Se trata de una arquitectura que permite la cooperación entre diferentes sistemas (módulos de comprensión, reconocedores, gestores de diálogo, etc. desarrollados para diferentes entornos de uso) de un modo fácil y flexible para el desarrollo de sistemas distribuidos de diálogo hablado aunque es posible su extensión a cualquier otro tipo de tarea. Se ha presentado su interfaz de comunicaciones multiplataforma y descrito sus elementos constitutivos, procedimientos de gestión, protocolos y algunos de los servicios desarrollados.

Los autores desean agradecer a todos los investigadores del proyecto coordinado EDECAN su colaboración y aportaciones a este trabajo.

#### 7. BIBLIOGRAFÍA

- [1] Seneff, S. et al. "Galaxy-II: a reference architecture for conversational systems development". ICSLP, 931-934. 1998.
- [2] V. Zue, et al., "JUPITER: A Telephone-Based Conversational Interface for Weather Information," IEEE Trans. on Speech and Audio Proc., Vol. 8, No. 1, Jan. 2000
- [3] Walker M., Hirschmann L., Aberdeen J. "Evaluation for DARPA COMMUNICATOR Spoken Dialog Systems". LREC, 2000.
- [4] Hacioglu, K. and Pellom, B. "A Distributed Architecture for Robust Automatic Speech Recognition". ICASSP, 328-331. 2003.
- [5] L. Buera, E. Lleida, A. Miguel, A. Ortega, O. Saz, "Cepstral vector normalization based on stereo data for robust speech recognition", IEEE trans on Audio, Speech and Language Processing, vol.15, pp.1098-1113. Marzo 2007.