Generación semiautomática de aplicaciones de diálogo multimodales: Proyecto GEMINI

R. Córdoba, L.F. D'Haro, J.M. Montero, J. Ferreiros, J. Macías-Guarasa, J.D. Romeral, J.M. Pardo

Grupo de Tecnología del Habla. Departamento de Ingeniería Electrónica. Universidad Politécnica de Madrid
E.T.S.I. Telecomunicación. Ciudad Universitaria s/n, 28040-Madrid, Spain
Telf: 91 5495700 ext. 343, Fax: 91 3367323
E-mail: cordoba@die.upm.es
http://www-gth.die.upm.es

Resumen

Presentamos en esta ponencia una herramienta de generación semiautomática de aplicaciones de diálogo hombre-máquina en la que, partiendo únicamente una descripción de la base de datos de un servicio y tras una interacción mínima con el diseñador, se generan simultáneamente diálogos en varios idiomas y dos modalidades, voz y web. Se demuestra la eficacia del sistema en el desarrollo de una aplicación bancaria en la que el usuario puede obtener información genérica de productos del banco, de su cuenta, hacer transferencias, etc. Así mismo, se presenta la tecnología necesaria para que la plataforma en tiempo real en la modalidad de voz, que aprovecha los diálogos generados por la herramienta, funcione de la forma más satisfactoria posible, minimizando el tiempo necesario para la consulta del usuario.

Palabras claves: sistemas automáticos de diálogo, multimodalidad, multilingualidad, reconocimiento de habla, VoiceXML.

1. Introducción

El proyecto Gemini (Generic Environment for Multilingual Interactive Natural Interfaces, IST-2001-32343), objeto de esta comunicación (Gemini 03), es un proyecto de dos años (2002-2004), financiado por la Unión Europea en el que interviene nuestro grupo representando a la UPM.

El objetivo fundamental del proyecto, que se encuentra en su punto intermedio, es el desarrollo de un sistema de generación de aplicaciones de diálogo de forma semiautomática partiendo de una descripción de la base de datos. Con ese fin, se está desarrollando una herramienta mediante la cual el diseñador podrá especificar totalmente la aplicación de una forma rápida, flexible, intuitiva y cómoda, reduciendo drásticamente el tiempo necesario para su desarrollo.

Un sistema de diálogo se puede definir como cualquier sistema que ofrece de forma automática un servicio al usuario (contenido en una base de datos) mediante un diálogo hombre-máquina, estando la máquina en una ubicación remota. Como ejemplo de servicio podemos mencionar los datos de un cliente del banco para una aplicación bancaria, los datos de todos los viajes posibles en una aplicación de agencia de viajes, etc.

Este diálogo se puede hacer mediante varias modalidades: voz por teléfono (mediante técnicas de reconocimiento de habla y conversión texto-voz), a través de Internet, PDA, WAP, etc. Se habla en estos casos de multimodalidad, que es uno de los objetivos del proyecto: conseguir que, con un mismo diseño de la aplicación, nuestra plataforma genere simultáneamente el código para ofrecer el servicio, en este caso, a través del teléfono mediante voz y/o a través de Internet.

Otro de los grandes objetivos del proyecto es la multilingualidad. De nuevo, a partir de un mismo diseño de la aplicación, el sistema debe generar en paralelo varios diálogos para atender a usuarios de países distintos. En el proyecto, estamos trabajando en cuatro idiomas: alemán, griego, español e inglés.

Por lo tanto, nos dirigimos a dos tipos de población objetivo interesadas. Por un lado, los proveedores de servicios, que son los que van a utilizar la herramienta de generación de diálogos para ofrecer sus servicios. Para ellos, la mayor ventaja del proyecto es la reducción de tiempo que les supone el poder utilizar la herramienta, en vez de tener que diseñar la aplicación desde cero. Por otra parte, están los usuarios del servicio, que, gracias a la reducción de costes, podrán acceder cada vez más a un mayor número de servicios las 24 horas del día sin necesidad de operadores.

Lógicamente, como parte importante del proyecto se encuentra el poder demostrar la eficacia de la herramienta de generación de diálogos. Para ello, se va a utilizar dicha herramienta o sistema de desarrollo para generar dos aplicaciones que se van a probar en entornos de funcionamiento reales: una aplicación bancaria, que permite un gran número de operaciones al usuario, y una aplicación de atención al ciudadano, donde se le proporciona información al ciudadano acerca de trámites administrativos, como renovar el DNI, etc.

A lo largo del artículo, llamaremos usuario al cliente final de nuestro sistema, y diseñador a la persona que construye el sistema de diálogo. Nos centramos fundamentalmente en el desarrollo de la herramienta de generación de diálogos, en un asistente concreto de la herramienta y en la aplicación bancaria que se ha implementado utilizando esta plataforma, dado que es la que ha desarrollado nuestro grupo dentro del consorcio.

2. Antecedentes

Este proyecto es la continuación del proyecto IDAS (LE4-8315), que ya presentamos en estas jornadas en años precedentes (San-Segundo 99, Córdoba 00), obteniendo en ambos casos un premio a la mejor ponencia en sus categorías respectivas. Los resultados de dicho proyecto se presentaron también en congresos internacionales relevantes como (Córdoba 01, Córdoba 02, Lehtinen 00.)

En dicho proyecto, se desarrolló un demostrador capaz de dar un servicio de páginas blancas por teléfono en el que se proporcionaba a los usuarios números de teléfono (o fax) de particulares y de empresas. Fue evaluado con usuarios reales y sus tasas de éxito fueron muy positivas.

El proyecto actual es claramente más ambicioso, dado que no sólo cubre la realización de una plataforma telefónica con la que proporcionar el servicio de la aplicación bancaria, sino que también busca la generación semiautomática de diálogos para múltiples idiomas y múltiples modalidades.

3. La herramienta de generación de diálogos (AGP)

Se ha creado una herramienta de generación de diálogos llamada AGP (Application Generation Platform, Plataforma de generación de aplicaciones) que es un conjunto integrado de herramientas y asistentes con los que automatizar el diseño de aplicaciones de diálogo. La arquitectura trata de conseguir los siguientes **objetivos fundamentales**:

- La participación del diseñador debe ser mínima o muy reducida si la comparamos con la utilización de otras herramientas. Para ello, la interacción sistema-diseñador se basa en una serie de asistentes que automatizan y simplifican el trabajo, guiando al diseñador hacia su objetivo.
- Se busca la utilización de estándares para que la herramienta sea "autónoma", es decir, que sea independiente de las plataformas utilizadas en el proyecto. Por ese motivo, el resultado de la herramienta para la modalidad de voz es un guión generado de acuerdo al estándar VoiceXML para el que existen numerosos intérpretes y plataformas comerciales capaces de utilizarlos. Del mismo modo, el resultado para la modalidad de Web es un guión en xHTML, por lo que cualquier navegador puede aprovecharlos, además de conseguir la independencia del sistema operativo.
- Es necesario separar los componentes del diálogo que son dependientes de la modalidad o del idioma de los componentes que son independientes.
- Se deben proporcionar al diseñador una serie de bibliotecas o módulos con los que resolver situaciones cotidianas. Por ejemplo, tratamiento de errores, obtención de series de números, de fechas, corrección de errores del reconocimiento, verificación de lo reconocido, etc.

Así mismo, es importante destacar que se ha diseñado dentro del proyecto una sintaxis basada en XML que hemos llamado GDialogXML (Gemini Dialog XML). Es un lenguaje de modelado abstracto orientado a objetos. Todos los asistentes de la arquitectura generan su salida siguiendo dicha sintaxis, lo que simplifica la interacción entre los mismos y, de este modo, facilita la división del trabajo entre los socios del proyecto. Al diseño de esta sintaxis se le ha dedicado una gran atención y ha sido revisada en numerosas ocasiones.

En la Figura 1 puede observarse la **arquitectura** elegida para el AGP.

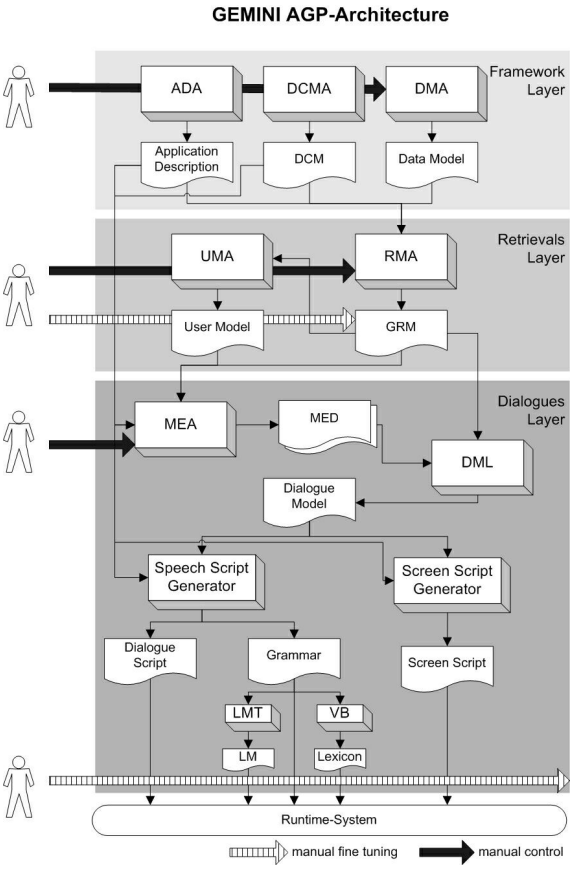

Figura 1. Diagrama esquemático de la arquitectura del AGP.

La arquitectura consta de **tres niveles básicos** e independientes:

1) Nivel superior (Framework Layer). Aquí, el diseñador debe especificar todos los aspectos globales relacionados con la aplicación y los datos que va a utilizar la misma. Tiene 3 partes:
   - Asistente de descripción de la aplicación (ADA): se definen las características básicas de la aplicación, como los idiomas, modalidades, bibliotecas a utilizar, etc.
   - Asistente del modelo de los datos (DMA): se introducen las descripciones de las clases de la base de datos en que se apoya el servicio, así como sus atributos, claves primarias, etc.
   - Asistente de conexión con el modelo de datos (DCMA): es donde se definen las funciones de acceso a la base de datos específica. Su objetivo es independizar la base de datos de la aplicación. La aplicación llama a funciones definidas en este asistente. Por ejemplo, la función getPersonaDadoDNI se definiría e implementaría aquí y se podría utilizar en el sistema en tiempo real sin necesidad de conocer las características de la base de datos. En este módulo trabajaría un experto en bases de datos más que el diseñador del diálogo.
2) Nivel intermedio (Retrievals Layer). Es donde se define el diálogo de una forma independiente del idioma y de la modalidad. Consta de dos módulos:
   - Asistente de modelado de la recuperación (RMA). Se describen los estados de diálogo que componen el servicio, las interacciones con el usuario, las transiciones entre los estados del diálogo, etc.
   - Asistente de modelado de usuario (UMA): puede consultarse en el apartado 5.7 el objetivo del modelado de usuario. Aquí es donde se introducen sus opciones.
3) Nivel inferior (Dialogues Layer), donde se completa el diálogo introduciendo los aspectos dependientes del idioma y de la modalidad. A continuación, el sistema genera automáticamente los guiones de la aplicación en todas las modalidades definidas. Consta de los módulos siguientes:
   - Asistente de extensión de modalidad (MEA): se describen los aspectos dependientes del idioma (frases que pronuncia el sistema, vocabularios y gramáticas utilizados en cada idioma, etc.) y de la modalidad (los modos de confirmación, el manejo de errores, cuántos resultados se presentan simultáneamente al usuario, etc., que son todos ellos diferentes entre la modalidad de voz y la de web). Como puede observarse en la Figura 1, su resultado (llamado MED, Descripción de extensión de modalidad) se une al resultado del RMA (llamado GRM, Modelo genérico de recuperación), por lo que completa el esquema del flujo especificado en el RMA.
   - Vinculador del modelo de diálogo (DML): simplemente une el resultado del RMA y del MEA para formar el modelo de diálogo completo. No hay interacción con el usuario.
   - A continuación, se generan automáticamente los guiones de ejecución en el estándar VoiceXML para voz y en xHTML para web.
   - Hay dos herramientas adicionales en la arquitectura: la Herramienta de modelado de lenguaje (LMT), donde se generan los modelos de lenguaje a utilizar en el reconocimiento (aporta la información lingüística al reconocedor, véase el apartado 5.4), y la Herramienta de creación de vocabularios (VB), con la que se decide el vocabulario (las palabras permitidas en cada etapa) a utilizar en el reconocimiento.

4. Un asistente concreto: el RMA

El RMA es un asistente clave en el proyecto, al ser el primer sitio en el que se definen los pasos de los que consta el servicio. Además, ha sido desarrollado íntegramente en nuestro grupo.

Para que el sistema sea productivo para el diseñador, es necesario que este asistente sea lo más intuitivo posible y que automatice el diseño del servicio, ofreciendo las ayudas oportunas. Un aspecto importante en esta estrategia es proporcionar bibliotecas con diálogos diseñados anteriormente para objetos de datos similares. Por ejemplo, fechas, ciudades, información basada en dígitos, etc.

En la Figura 2 se puede observar la pantalla principal del asistente al comenzar a editar el servicio. Sin entrar en demasiados detalles, se le ofrecen automáticamente al diseñador una serie de diálogos: bienvenida al sistema, despedida, un conjunto de diálogos de obtención de información preguntando al usuario (“Ask for information” dialogs) y de proporcionar información al usuario (“Show information” dialogs) basados en la información del Modelo de datos, es decir, las clases y atributos de los datos del servicio; y un conjunto de diálogos obtenidos de las bibliotecas que se hayan definido en la descripción de la aplicación. Esos diálogos se pueden aprovechar en todo momento mediante “arrastrar y soltar”.

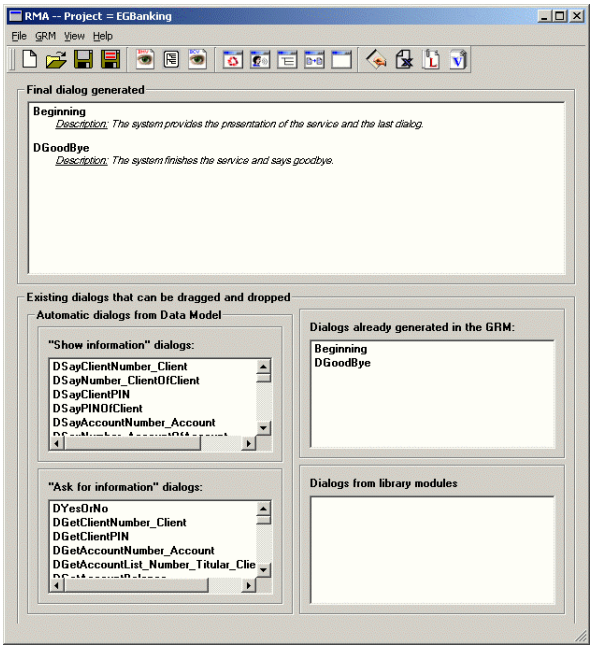

Figura 2. Pantalla principal del RMA.

El usuario puede añadir todo tipo de diálogos con los que configurar el servicio. Hemos considerado cinco tipos básicos: basados en un bucle, en una secuencia de acciones (o subdiálogos), en información introducida por el usuario, en el valor de una variable y en blanco (pensado para permitir la llamada a un diálogo que se va a definir posteriormente.)

En función del tipo de diálogo elegido, el diseñador tendrá que introducir una serie de datos diferente. En la Figura 3 podemos ver un ejemplo de estas pantallas, en este caso dedicada a introducir un bucle en el servicio. Existe además la posibilidad de definir y utilizar variables locales y globales, definir las variables de entrada y salida en cada diálogo (como si fueran subrutinas), insertar llamadas a otros diálogos o funciones de acceso a datos (consultas definidas en el DCMA), introducir estructuras if-then-else dentro de los diálogos, etc. De ahí la existencia de tantas opciones en la pantalla de la Figura 3.

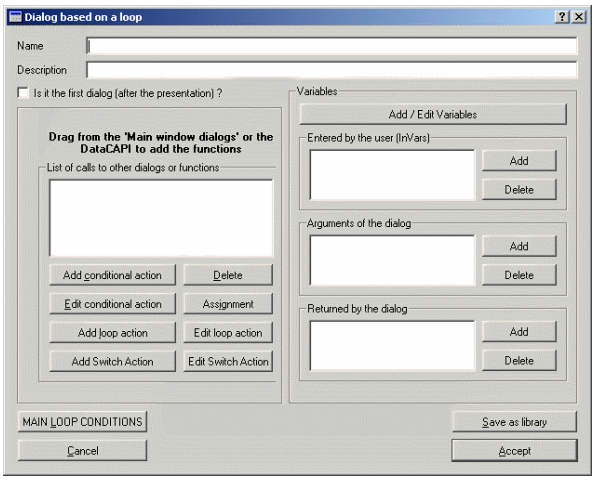

Figura 3. Pantalla en la que se define un diálogo basado en un bucle de acciones.

5. La tecnología subyacente

Para poder implementar esta tecnología en la modalidad de voz es necesario disponer de una serie de módulos básicos. Son los siguientes:

5.1. Módulo de reconocimiento

Es probablemente el módulo clave en un sistema de diálogo por teléfono, dado que los fallos en este módulo afectan drásticamente a la interacción hombre-máquina, obligando a introducir confirmaciones, repeticiones por parte del usuario, etc. Por lo tanto, se ha hecho un gran esfuerzo en mejorar dicho módulo con respecto al utilizado en el proyecto IDAS.

En la actualidad, tenemos un sistema de reconocimiento que utiliza modelos ocultos de Markov (HMM) continuos entrenados con la base de datos SpeechDat, de 4.000 locutores, que dispone de unas 46 horas de grabación de habla continua. El sistema utiliza modelos contextuales agrupados mediante un algoritmo de árboles de decisión. El sistema tiene 1807 estados diferentes, con 6 gaussianas por estado. La tasa de error, para una tarea de habla continua por teléfono de 3.065 palabras, es de únicamente un 4.2%.

Además, en nuestro sistema utilizamos un reconocedor de habla aislada de características similares al anterior para los casos en que se espere que el usuario responda con una palabra aislada. Lógicamente, la tasa de error es menor cuando se tienen que detectar palabras aisladas. También utilizaremos en nuestro sistema reconocedores adaptados al reconocimiento de fechas, reconocimiento de dígitos y deletreo (se pasa a deletreo cuando hay errores en el reconocimiento.)

Además, se ha hecho un gran esfuerzo en adaptar nuestro software de reconocimiento al intérprete de VoiceXML llamado OpenVXI. Para la modalidad de habla, el resultado del AGP es, como ya hemos comentado, un guión del diálogo escrito en lenguaje VoiceXML. Para poder ejecutarlo, necesitamos de un intérprete. Hemos elegido OpenVXI por ser una solución de código abierto y por su portabilidad, no requiere de ningún motor de reconocimiento o síntesis de voz en particular, ni es específico para una plataforma telefónica concreta. Como inconvenientes cabe destacar la escasez de una documentación válida, lo que dificulta la introducción de los motores en el intérprete y la ausencia de una implementación de referencia con funcionalidad real.

5.2. Módulo de conversión texto-voz

Para ofrecer al usuario mensajes de contenido variable sin necesidad de grabarlos previamente. Estamos utilizando el conversor desarrollado por el grupo, ampliamente probado y distribuido comercialmente (Pardo 95.)

5.3. Modelado de diálogo

Es el módulo básico del proyecto, donde modelamos las interacciones hombre-máquina, buscando que sean rápidas e intuitivas.

En este módulo se debe identificar los objetivos que desea el usuario y, en función de ellos (por ejemplo, reservar un billete de avión) decidir qué pasos se deben seguir en el diálogo hasta lograrlos. Hay tres tipos de estrategias:

- De iniciativa del sistema, donde el sistema lleva toda la iniciativa, haciendo todas las preguntas, y el usuario se limita a contestar a las preguntas sin aportar información adicional.
- De iniciativa del usuario: es el usuario el que lleva la batuta. Son sistemas muy ambiciosos porque un fallo en el sistema de reconocimiento puede tener consecuencias desastrosas si no se averigua lo que desea realmente el usuario. Obviamente es la meta a la que llegar, dado que son los que proporcionan mayor naturalidad y se necesita menos tiempo en completar la consulta.
- De iniciativa mixta: es una solución intermedia. El sistema hace las primeras preguntas pero el usuario puede introducir información adicional y el sistema aprovecharla. Por ejemplo, el sistema puede preguntar "¿adónde desea viajar?" Y el usuario podría contestar "A Barcelona el lunes que viene".

En el proyecto hemos comenzado con los diálogos con iniciativa del sistema, dadas las dificultades a las que nos hemos tenido que enfrentar con la multimodalidad y multilingualidad dentro de nuestra arquitectura de diseño. En el segundo año del proyecto abordaremos el problema de la iniciativa mixta.

5.4. Modelado del lenguaje

Modela las posibles combinaciones de palabras que se pueden producir en cada paso del diálogo (en el caso de gramáticas estocásticas, modelamos la probabilidad de que una palabra siga a otra), con lo que esta información ayuda al módulo de reconocimiento a incrementar su tasa de aciertos.

Lo más difícil es utilizar modelos de lenguaje específicos de una etapa del diálogo obtenidos a partir de un modelo de lenguaje genérico. La idea es entrenar un modelo genérico del habla utilizando una gran cantidad de datos. Como de la etapa del diálogo concreta disponemos de pocos datos, lo que hacemos es partir del modelo genérico y adaptarlo a la etapa del diálogo maximizando la verosimilitud de los datos disponibles en dicha etapa.

Se va a desarrollar un gran esfuerzo a esta técnica de adaptación. Disponemos en el grupo de modelos de lenguaje genéricos del castellano, por lo que tienen un alto índice de confusabilidad (no guían demasiado al módulo de reconocimiento.) La idea básica es adaptar un modelo de lenguaje genérico a una situación específica (por ejemplo, preguntar una fecha para un viaje, una ciudad) utilizando un número reducido de frases ejemplo de esa situación.

5.5. Módulo de comprensión

Es un analizador semántico que detecta y representa semánticamente lo introducido por el usuario (los conceptos.) Basándose en esos conceptos el módulo de diálogo decide el paso siguiente de la interacción con el usuario.

5.6. Generación de lenguaje natural

Es la generación de texto a partir de representaciones semánticas abstractas (conceptos.) Se pasa de conceptos independientes de la modalidad y del idioma a las frases concretas a reproducir al usuario. El objetivo es parecer natural y no "aburrirle", teniendo en cuenta el nivel de experiencia del usuario (ver el modelado de usuario a continuación.)

5.7. Modelado de usuario

Consiste en introducir alternativas en el diálogo básico para adaptarse a la experiencia del usuario con el sistema. Se definen una serie de niveles de experiencia, y en función de ellos, el sistema ofrecerá más o menos información.

Por ejemplo, si el usuario es novato, se le proporcionará una ayuda más detallada, se le harán preguntas más concretas, se intentará confirmar lo que dice, etc.

Para decidir a qué nivel debe pertenecer un usuario, dado que en una llamada telefónica no se le suele identificar, hay que tener en cuenta una serie de factores como: rapidez en las respuestas, tasas de acierto del reconocimiento, solicitudes de ayuda, etc. (San-Segundo, 01).

5.8. Módulo de reconocimiento de idioma

Uno de los objetivos del sistema es ser multilingüe. Para ello, debemos detectar el idioma del usuario actual utilizando un pequeño fragmento de lo que dice inicialmente, tras lo cual conmutamos al sistema de reconocimiento del idioma detectado.

Se pueden utilizar distintas técnicas. En el grupo hemos utilizado una llamada PPRLM, que se basa en modelar información de la secuencia de fonemas que se obtiene para cada uno de los idiomas utilizando un reconocedor de fonemas muy simple.

Se pueden consultar los resultados en la discriminación entre inglés y castellano en (Córdoba 03.)

5.9. Módulo de reconocimiento de locutor

En algunas aplicaciones, se puede necesitar la identificación del usuario que llama (por ejemplo, en una aplicación bancaria.) Esa identificación puede ser manual, utilizando una serie de códigos, o automática, utilizando técnicas de reconocimiento de locutor. De este modo, se pueden utilizar modelos específicos del usuario actual, tanto acústicos como de lenguaje, lo que supone una mejora en la tasa de acierto del sistema y la mayor satisfacción del usuario.

6. La aplicación bancaria

Utilizando la herramienta de generación de diálogos hemos desarrollado un diálogo mediante el cual el usuario puede obtener todo tipo de información bancaria agrupada en varias categorías.

En la primera versión que acabamos de desarrollar se ofrece, mediante un diálogo con iniciativa del sistema:

- Información general de productos del banco: créditos personales, créditos para compra de coche, hipotecas; depósitos y sus tipos de interés; tarjetas de crédito / débito, etc.
- Autenticación del usuario: el usuario debe introducir su número de cuenta y un PIN.
- Consultas y transacciones para las cuentas del cliente: consulta de saldos y movimientos para las cuentas y tarjetas de crédito, realización de transferencias entre cuentas, etc.

En el momento de escribir esta ponencia todavía no se había probado la aplicación bancaria con usuarios reales, por lo que la evaluación de la misma formará parte de una comunicación futura.

7. Conclusiones

Se ha desarrollado un sistema de generación de diálogos que es extremadamente potente, capaz de generar de una forma semiautomática diálogos válidos para múltiples idiomas y dos modalidades partiendo de una descripción de la base de datos y de una interacción reducida con el diseñador.

El resultado es una plataforma potente y abierta con la que desarrollar diálogos de forma rápida y amigable. Así mismo, la utilización de estándares como VoiceXML y una sintaxis basada en XML nos coloca en una buena posición en el creciente mercado de los sistemas de diálogo basados en VoiceXML y en xHTML.

8. Referencias

[Córdoba 00] Córdoba, R., R. San-Segundo, J. Colás, J.M. Montero, J. Ferreiros, J. Macías-Guarasa, A. Gallardo, J.M. Gutiérrez, J.M. Pardo. "Optimización de un servicio automático de paginas blancas por teléfono: proyecto IDAS". X Jornadas TELECOM I+D. 2000.

[Córdoba 01] Córdoba, R., R. San-Segundo, J.M. Montero, J. Colás, J. Ferreiros, J. Macías-Guarasa, J.M. Pardo. "An Interactive Directory Assistance Service for Spanish with Large-Vocabulary Recognition", EUROSPEECH, pp. 1279-1282. 2001.

[Córdoba 02] Córdoba, R., J. Macías-Guarasa, J. Ferreiros, J.M. Montero, J.M. Pardo. "State Clustering Improvements for Continuous HMMs in a Spanish Large Vocabulary Recognition System", International Conference on Spoken Language Processing, pp. 677-680. 2002.

[Córdoba 03] R. Córdoba, G. Prime, J. Macías-Guarasa, J.M. Montero, J. Ferreiros, J.M. Pardo. "PPRLM Optimization for Language Identification in Air Traffic Control Tasks", EUROSPEECH, 2003.

[Gemini 03] Página web del proyecto Gemini: www.gemini-project.org.

[Lehtinen 00] Lehtinen, G., S. Safra, …, J.M. Pardo, R. Córdoba, R. San-Segundo, et al., "IDAS : Interactive Directory Assistance Service", VOTS-2000 Workshop, Belgium.

[Pardo 95] Pardo, J.M., et al "Spanish text to speech: from prosody to acoustic" International Conference on Acoustic 95 vol III, 1995.

[San-Segundo 99] San-Segundo, R., Colás, J., Montero, J.M., Córdoba, R., Ferreiros. J., Macías-Guarasa J., Gallardo A., Gutiérrez, J.M., Pastor, J., Pardo, J.M.: "Servidores vocales interactivos: desarrollo de un servicio de páginas blancas por teléfono con reconocimiento de voz - proyecto IDAS (interactive telephone-based directory assistance service). IX jornadas Telecom. I+D. 1999.

[San-Segundo 01] R. San-Segundo, J.M. Montero, J. Colás, J. Gutiérrez, J.M. Ramos, J.M. Pardo, "Methodology for Dialogue Design in Telephone-Based Spoken Dialogue Systems: a Spanish Train Information System", EUROSPEECH, pp. 2165-2168. 2001.