

# Nueva Técnica de Generación Automática de Gramáticas Para Sistemas de Diálogo

Zoraida Callejas, Ramón López-Cózar

Dpto. Lenguajes y Sistemas Informáticos, E.T.S. Ingeniería Informática  
18071 Universidad de Granada, Tel.: +34 958 240579, Fax: +34 958 243179  
E-mail: zoraida@correo.ugr.es, rlopezc@ugr.es

**Resumen:** Este artículo presenta un sistema de diálogo unimodal que permite consultar información académica en el entorno universitario. Nuestra principal aportación es el desarrollo de un módulo en el sistema que permite generar automáticamente gramáticas en formato JSGF que se utilizan en el proceso de reconocimiento automático del habla. El vocabulario que se incluye en las gramáticas es desconocido a priori, y se especifica mediante una interfaz web de fácil uso. Dicho vocabulario se encuentra almacenado en la base datos del sistema. La generación de las gramáticas es portable e independiente del contenido y estructura de la base de datos, por lo que la técnica propuesta puede ser de gran utilidad para los diseñadores de este tipo de sistemas.

**Palabras clave:** Sistemas de diálogo, reconocimiento y síntesis de habla, modelado del lenguaje, gramáticas, interacción persona-ordenador.

**Abstract:** This paper presents an unimodal dialogue system to query academic information in the University framework. Our main contribution is the implementation of a system module to create automatically the JSGF grammars that are used for automatic speech recognition. The vocabulary to include in the grammars is unknown a priori, and is specified by means of an easy-to-use web interface. Such a vocabulary is stored in the system database. The grammar generation is portable and independent from the database content and structure. Therefore, the proposed technique can be very useful for designers of dialogue systems.

**Keywords:** Dialogue systems, automatic speech recognition and synthesis, language modelling, grammars, human-computer interaction.

## 1 Introducción

El desarrollo de nuevas tecnologías en Informática y Telecomunicaciones ha favorecido que los usuarios puedan acceder a la información desde cualquier sitio y en cualquier momento. La aparición de Internet y su posterior expansión ha posibilitado que muchas empresas y entidades públicas se den a conocer a usuarios de todo el mundo a través de la red, a la que se puede acceder de forma móvil a través del sistema GPRS (General Packet Radio Service), tecnología móvil que proporciona altas velocidades de transferencia de datos. Próximamente, esta tecnología será sustituida por la tercera

generación de redes móviles, UMTS (Universal Mobile Telecommunications System), la cual permitirá mayor capacidad para el uso de aplicaciones como correo electrónico o mensajería instantánea.

El teléfono se ha convertido en un dispositivo indispensable para muchos usuarios (Huuskonen et al. 2003). No obstante, el acceso a la red a través de teléfonos móviles no es una práctica socialmente extendida, ni de tanto calado como el acceso a través de un ordenador. Por otra parte, la navegación entre páginas web se realiza hoy en día principalmente de forma visual, lo cual impide el acceso a la red a personas con discapacidad visual (Varela 2004).

Con la aparición de los denominados *sistemas de diálogo* surge un nuevo paradigma en la comunicación entre personas y ordenadores que ha suscitado un

---

Este trabajo ha sido subvencionado por el Plan Propio de Investigación de la Universidad de Granada, 2005.

gran interés entre la comunidad científica (Llisterri 2004). La interacción oral es la que más comúnmente usamos los seres humanos para comunicarnos y el hecho comunicativo surge muchas veces con la necesidad de dar a conocer o consultar información que un interlocutor conoce y el otro no.

En este artículo presentamos un nuevo sistema de diálogo unimodal llamado UAH (Universidad Al Habla) que está siendo desarrollado en nuestro laboratorio con objeto de permitir el acceso oral (a través del teléfono) a la información disponible en la página web de nuestra Universidad. Gracias a este sistema se posibilita el uso de ambas modalidades de acceso (visual y oral), lo que permite que usuarios con discapacidad visual o auditiva puedan acceder a la información, eligiendo la modalidad que más se adapte a sus necesidades.

Para implementar el sistema estamos utilizando el lenguaje de marcas VoiceXML, por diversas razones. En primer lugar, el lenguaje aporta flexibilidad e independencia de la máquina en la que se realiza o se ejecuta el programa en desarrollo. En segundo lugar, existe una especificación completa de las versiones 1.0 y 2.0 en el Consorcio Web W3C que ha convertido al lenguaje en un estándar, proveyendo de un foro donde miles de desarrolladores de aplicaciones VoiceXML comparten sus experiencias. Además, existen numerosas herramientas y sistemas de desarrollo que permiten realizar la implementación de una forma cómoda (en nuestro caso, estamos usando un toolkit desarrollado por una empresa integrada en el mencionado consorcio). Por último, VoiceXML puede usarse junto con lenguajes de programación de servidores web, como PHP, JSP o ASP, los cuales permiten realizar programas dinámicos que se ajusten a las necesidades del usuario en tiempo de ejecución.

La principal novedad de nuestro sistema es un módulo que permite crear automáticamente gramáticas JSGF (Java Speech Grammar Format) que son usadas durante el proceso de reconocimiento automático del habla (RAH). El vocabulario a incluir en las mismas es desconocido a priori, pues se extrae de una base de datos.

En nuestro sistema, este vocabulario incluye entre otras palabras, los nombres y apellidos de todos los profesores de nuestra Universidad, así como el de las asignaturas que imparten.

El artículo está estructurado de la siguiente forma. La sección 2 presenta una visión general de la arquitectura del sistema en la que se describen los principales módulos. La sección 3 se centra en el módulo de generación automática de gramáticas propuesta, su motivación, estructura y funcionamiento. Por último, la sección 4 presenta las conclusiones y comenta algunas líneas de trabajo futuro.

## 2 Arquitectura del sistema

La Figura 1 muestra la arquitectura modular de nuestro sistema de diálogo. Como se puede observar, el sistema se compone de los cinco módulos típicos de los sistemas de diálogo actuales, dedicados a las siguientes funciones: RAH, gestión del diálogo (GD), acceso a la base de datos (Acceso a BD), almacenamiento de datos (BD) y generación de respuesta oral (GRO).

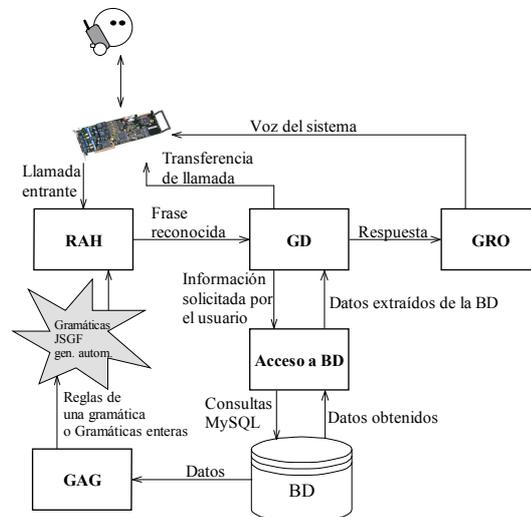


Figura 1. Arquitectura del sistema de diálogo

El módulo de generación automática de gramáticas (GAG), que implementa la técnica propuesta, se describe en la sección 3. Describimos a continuación, de forma resumida, el funcionamiento de los restantes módulos.

## 2.1 Reconocimiento Automático del Habla

El módulo de RAH se encarga de procesar cada frase del usuario (señal de voz proveniente de la tarjeta de interfaz telefónica Dialogic D/41JCT-LS) y transformarla en una secuencia de palabras en formato de texto. Este proceso se lleva a cabo utilizando catorce gramáticas JSGF. Como se comenta en la sección 3, algunas de estas gramáticas se generan de forma manual y otras de forma automática mediante la técnica propuesta en este artículo, la cual se describe en el apartado 3.2. Las gramáticas JSGF adoptan el estilo del lenguaje de programación Java, lo que las hace independientes de la plataforma y del fabricante. Su uso junto con lenguajes de programación como VoiceXML permite obtener aplicaciones basadas en voz de gran flexibilidad (Hartl 2003).

Este módulo, implementado mediante un sistema de RAH comercial, admite *barge-in*, es decir, permite que el usuario pueda interrumpir la salida oral del sistema cuando lo desee pronunciando una nueva frase, que es procesada por el reconocedor. Este mecanismo permite mejorar la interacción del usuario, pero a su vez, como indican Ström y Seneff (2000), introduce nuevas dificultades, puesto que el usuario no habrá escuchado la salida del sistema en su totalidad, lo que habrá que tener presente en la posterior etapa de gestión del diálogo.

## 2.2 Gestión del diálogo

El gestor del diálogo (GD) se encarga de determinar la siguiente respuesta a proporcionar al usuario en función de los datos recabados, así como de guiar al usuario durante la interacción. La complejidad de este módulo depende de varios factores. Por una parte, de lo intrincado de la interacción que se quiera modelar. Por otra, del tipo de tarea que se quiera resolver mediante el diálogo. Finalmente, también depende de la flexibilidad deseada para el diálogo y del tipo de iniciativa que se desee implementar (dirigida por el usuario, por el sistema o mixta). En nuestro caso, la tarea es la extracción de información en el entorno Universitario, estando la iniciativa dirigida

por el sistema en los estados del diálogo denominados de *elección básica*. Asimismo, nuestro sistema usa iniciativa mixta en los estados denominados de *elección de detalles*.

En los estados de *elección básica* el usuario determina el área de la que desea obtener información (p. e. acerca de un profesor o una asignatura). En este caso, la iniciativa en el diálogo está dirigida por el sistema para que el usuario pueda obtener cómoda y rápidamente la información que desea, de entre los diversos tipos de información que el sistema le ofrece.

En los estados de *elección de detalles* el usuario identifica el tipo de información concreto que desea conocer dentro de una determinada área (p. e. el número de despacho de un profesor). El uso de iniciativa mixta en estos estados favorece que el usuario disfrute de suficiente flexibilidad para expresarse, una vez determinada el área (Glass y Seneff 2003).

Si bien no existe un punto de vista único sobre las funciones que un gestor del diálogo debe desempeñar, una de las visiones más ampliamente aceptadas es la presentada por Traum y Larsson (2001). En ella se expone que el gestor debe actualizar el contexto del diálogo para poder realizar las interpretaciones semánticas de las frases del usuario en un contexto dado. Además, debe procesar tareas en un determinado dominio (en nuestro caso, el contexto universitario) y debe poder decidir qué información expresar, cuándo hacerlo y cómo hacerlo.

En nuestro sistema, el gestor del diálogo adapta las respuestas dinámicamente al contexto y al estado del diálogo, expresando de distinta forma algunas frases para favorecer la naturalidad de la interacción. Por ejemplo, el contenido de la ayuda que proporciona al usuario se adapta al instante y al aspecto concreto del que se está hablando en cada momento de la interacción.

El contexto también se usa para determinar qué estrategia de confirmación utilizar. Por una parte, el sistema utiliza confirmaciones *explícitas* para las acciones más importantes, o para aquellas situaciones en las que el RAH sea más dificultoso. Por ejemplo, antes de transferir una llamada a un profesor se confirma

explícitamente el nombre del profesor (p. e. ¿Ha dicho profesor X?). En el resto de situaciones el sistema utiliza confirmaciones *implícitas*, pues de esta forma la interacción resulta más natural para el usuario (Nöth et al. 2004).

### 2.3 Acceso a BD

En la literatura podemos encontrar múltiples referencias a la importancia de separar, por una parte, las actividades de acceso y consulta a la base de conocimiento, y por otra, el resto del sistema del diálogo. Por ejemplo en el marco del proyecto GEMINI (D’Haro et al. 2004), se construyó un asistente de conexión con el modelo de datos que permite a los usuarios crear sistemas de diálogo de forma semiautomática con independencia de las características de la base de datos usada. En nuestro caso, proponemos una dicotomía entre la gestión del diálogo y el acceso a la información, mediante la creación de un módulo de acceso a la base de datos. El gestor del diálogo proporciona a este módulo la información que el usuario desea consultar, siendo este último el que construye la consulta MySQL correspondiente y extrae la información de la base de datos.

Una vez extraída, el módulo de acceso a la base de datos ejecuta un programa PHP que se encarga de comprobar la validez de los datos (si es que realmente se ha podido extraer alguna información de la base de datos). Asimismo, comprueba que no existan datos repetidos en el resultado antes de enviarlos al gestor del diálogo para que éste decida de qué forma comunicarlos al usuario.

Al igual que ocurre en el caso del sistema SAPLEN (López-Cózar y Rubio 1997, López-Cózar et al. 2000), en nuestro sistema en desarrollo se realizan dos tipos de consultas a la base de datos: *explícitas* e *implícitas*. En el primer caso, la consulta se realiza a iniciativa del usuario, mientras que en el segundo se realiza por iniciativa del sistema. Por ejemplo, si el usuario consulta el número de teléfono del profesor “Francisco Pérez” se realizan dos consultas. En la primera se comprueba el número de registros obtenidos de la base de datos a partir de esta condición (es decir,

nombreApellidos=“Francisco Pérez”). Si este número es cero, se informa al usuario que no existe ningún profesor con el nombre indicado. Si el número es mayor que uno, se solicita al usuario que seleccione el registro deseado. Una vez determinado el nombre, en su caso, se realiza otra consulta a la base de datos para obtener el número de teléfono solicitado. La primera consulta es *implícita* pues la realiza el sistema sin que exista una petición explícita del usuario. La segunda es una consulta *explícita* pues el usuario solicita al sistema que la realice.

### 2.4 Base de datos

El sistema obtiene la información solicitada por el usuario consultando una base de datos construida mediante MySQL, que contiene información pública de nuestra Universidad. Por ejemplo, el número de fax o la dirección de e-mail de los profesores, pero no sus datos personales o dirección postal. La base de datos ha sido diseñada con el fin de recoger datos de los departamentos que éstos suelen incluir en sus respectivas páginas web, aunque no se encuentren almacenados en ninguna base de datos. Nuestro sistema podría trabajar con una base de datos que almacenara todo tipo de información, creando vistas sobre ésta que permitieran mantener la privacidad de datos económicos y personales.

La base de datos utilizada es de tipo relacional y consta de 22 tablas que han sido obtenidas a partir de un diagrama entidad-relación, seguido de un proceso de fusión y normalización. A modo de ejemplo, la Figura 2 muestra los datos de la tabla “departamento”.

Estructura de la tabla <b>departamento</b> de la base de datos universidad					
Nombre	Tipo	Null	Indice	Por Defecto	Extras
Cod_depar	int(10) unsigned		PRI		auto_increment
Nombre	varchar(45)				
Telefono	decimal(9,0)	YES			
Fax	decimal(9,0)	YES			
Email	text	YES			
Direccion	varchar(45)	YES			
Web1	text	YES			
Web2	text	YES			

Figura 2. Tabla “departamento” de la BD

## 2.5 Generación de respuesta oral

La generación de la respuesta oral del sistema se realiza, en primera instancia, en formato de texto, utilizando 56 patrones que hemos clasificado en distintas categorías: *profesor*, *grupos de investigación*, *doctorado*, *departamento*, *asignaturas*, *titulación*, *confirmación*, *saludos* y *ayuda*. Una vez obtenida la respuesta en formato de texto, ésta se transforma a voz mediante un sistema de conversión texto-habla comercial.

Dentro de cada categoría de respuestas existe un patrón específico para cada dato que se pueda proporcionar al usuario en referencia a la misma. Por ejemplo, la categoría *profesor* tiene siete patrones (número de teléfono, horario de tutorías, asignaturas impartidas, dirección de su página web, dirección de las asignaturas que imparte, ubicación y número de su despacho, y dirección de su correo electrónico).

Para la categoría correspondiente a los grupos de investigación se han creado dos patrones (dirección de su página web y nombre del profesor responsable).

Para los cursos de doctorado se han diseñado patrones para: nombre del programa, mención de calidad, cursos que ofrece (nombre, tipo, carácter y número de créditos), nombre del coordinador, profesores que colaboran, y nombres de las titulaciones desde las que se puede acceder.

Para los despachos se ha creado patrones para proporcionar la dirección postal y la electrónica, el número de teléfono y el número de fax.

Las asignaturas cuentan con catorce patrones correspondientes al nombre, tipo, curso, página web, titulación, profesor responsable, profesores que la imparten, cuatrimestre en que se imparte, grupos que la componen, carácter de cada grupo, nombre de los grupos, horario y aula de los grupos, y profesores que imparten cada grupo específico.

Para la titulación tenemos los siguientes patrones: nombre, tipo, duración y facultad en que se imparte.

Por último, se han diseñado varios patrones para generar saludos, mensajes de ayuda y confirmaciones.

Cada patrón puede hacer uso de otros patrones de distintas categorías. Por ejemplo, para proporcionar la dirección de e-mail del profesor responsable de una determinada asignatura, se usan patrones de dos categorías: asignatura y profesor. Además, los patrones se adaptan a los cambios de género y número, generando por ejemplo los mensajes “El correo electrónico del profesor D. A es A@ugr.es” o “El correo de la profesora Dña. B es B@ugr.es”.

## 3 Técnica de generación automática de gramáticas (GAG)

El vocabulario que el sistema debe reconocer puede dividirse en cuatro grandes grupos. El primero, al que hemos denominado *palabras clave*, está compuesto por aquellas palabras o grupos de palabras que identifican una tupla de la base de datos, como por ejemplo el nombre de una asignatura o el de un profesor.

En segundo lugar, distinguimos las *palabras clase* que identifican tablas de la base de datos, por ejemplo “asignatura” o “profesor”, o campos en una tabla como por ejemplo “página web” o “número de créditos”.

En tercer lugar, encontramos las palabras y estructuras sintácticas que el usuario emplea para hacer construcciones (frases) que contengan palabras clave y/o palabras clase para solicitar información al sistema. Por ejemplo, en la frase “¿En qué aula se imparte Ingeniería del Software I?” la palabra *aula* es una palabra clase, mientras que “Ingeniería del Software I” es un conjunto de palabras clave.

Por último, distinguimos las palabras y estructuras que no suponen ninguna solicitud de información acerca de la Universidad, como por ejemplo las expresiones de saludo y despedida, las afirmaciones o negaciones, las expresiones de cortesía (como “por favor” o “gracias”) o las solicitudes de ayuda.

Las palabras y estructuras del tercer y cuarto tipo se pueden definir en la etapa de diseño del sistema, estableciendo qué tipo de construcciones se permitirán para solicitar ayuda, dar las gracias y saludar, así como para las distintas fórmulas de solicitud de información. El proceso de

RAH para estas construcciones se puede llevar a cabo empleando gramáticas de estados finitos estáticas (Rabiner y Juang 1993). Estas gramáticas especifican las expresiones válidas del usuario, devolviendo una secuencia de palabras reconocida, o bien, un conjunto de pares atributos-valor, según indica la especificación del W3C. Las palabras *clase* son los símbolos no terminales de las gramáticas de reconocimiento, mientras que las palabras *clave* son los símbolos terminales. Las palabras clave no son conocidas a priori, ya que se encuentran almacenadas en la base de datos.

En la siguiente sección se comentan diversas técnicas que permiten construir las gramáticas que se usarán en el proceso de RAH para estas construcciones, y cómo nuestra propuesta permite mejorar dicho proceso.

### 3.1 Técnicas existentes

En la literatura podemos encontrar diversas técnicas que abordan la tarea de construir gramáticas para sistemas de diálogo sin que el vocabulario sea conocido a priori, sino que debe ser consultado en una base de datos (como ocurre en nuestro caso). Un primer enfoque consiste en construir dinámicamente las gramáticas durante el proceso de RAH (Truillet et al. 2004). Esta técnica aporta flexibilidad ya que en todo momento se usa durante el RAH una gramática que tiene en cuenta los últimos cambios realizados en la base de datos. Sin embargo, este método puede suponer una carga computacional muy elevada, lo que se traduce en un incremento del tiempo de ejecución. En sistemas con bases de datos de cierta envergadura (p. e. información académica de una Universidad) este tiempo puede ser considerable, provocando que el sistema sea considerado “lento” por los usuarios.

Una segunda alternativa consiste en crear las gramáticas con anterioridad al proceso de RAH. Este es el caso de las gramáticas estáticas que se construyen como paso previo a la ejecución del sistema. Estas gramáticas se actualizan cada vez que éste se va a ejecutar, independientemente de que haya habido algún cambio en la base de datos

(Schalkwyck et al. 2003). Este método no supone un incremento del tiempo de RAH pero sí antes de éste. Por consiguiente, el incremento en la espera del usuario no se produce mientras se reconoce la frase pronunciada, sino cuando el sistema comienza a funcionar. Este incremento es menor desde el punto de vista del usuario pero sigue existiendo un retardo en la ejecución del sistema, que será mayor cuanto mayor sea el tamaño de la base de datos.

### 3.2 Técnica propuesta

Para evitar este problema, la técnica que proponemos consiste en usar gramáticas estáticas que son generadas de forma automática mediante un proceso dividido en dos etapas:

- i) Extracción de información de la base de datos.
- ii) Construcción de las gramáticas correspondientes a la información extraída.

Dichas gramáticas se generan con anterioridad al proceso de RAH, y son actualizadas conforme se produce algún cambio en los datos almacenados en la base de datos. La generación se realiza mediante el módulo GAG del sistema, implementado mediante PHP y MySQL, al que se accede mediante una interfaz web de fácil uso.

La principal ventaja de la técnica propuesta frente a otras que podemos encontrar en la literatura (p. e. ver Mc Tear 2004 y Buntschuh et al. 1998), radica en que es general, y por consiguiente, aplicable a la hora de generar gramáticas para cualquier sistema de diálogo que deba extraer información de una base de datos.

Como se puede observar en la Figura 3, el módulo de GAG solicita inicialmente la introducción del nombre de la base de datos del sistema de diálogo y de la tabla dentro de ella, así como el número de campos de la tabla desde los que extraerá la información. Asimismo, solicita el nombre que se dará al elemento (regla) en la gramática, y el nombre del fichero donde se almacenará la regla, bien en un fichero nuevo, o bien en uno ya existente que contiene una gramática. En el ejemplo de la Figura 3 se especifica que se usen dos campos de la

tabla “profesor” de la base de datos “Universidad”. El nombre que se da a la regla es “NombreProfesor”, la cual se almacena en el fichero “NombreProf.gram”. La Figura 4 muestra la selección de los dos campos de la tabla (“Nombre” y “PrimerApellido”). Finalmente, la Figura 5 muestra la gramática obtenida.

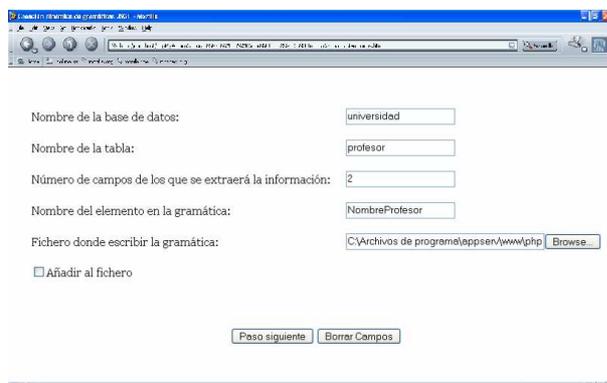


Figura 3. Introducción de datos iniciales para la generación de la gramática

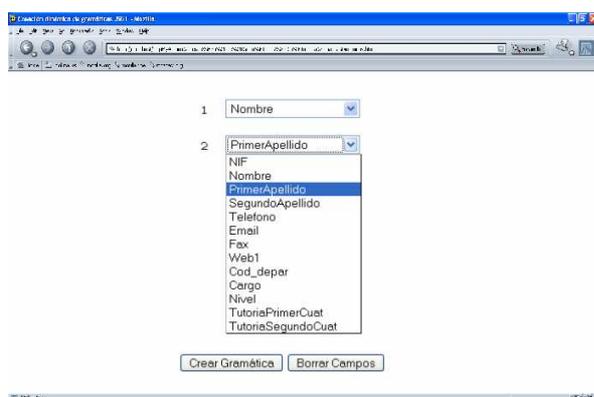


Figura 4. Elección de los campos de la tabla

```
#JSGF V1.0;
grammar NombreProfesor;

public <NombreProfesor>= (María del Mar
Abad | José Luis Garrido | Eladio Garví |
Miguel Gea | Francisco Blas Hernández | Juan
Antonio Holgado | Miguel Hornos | Ana
Anaya | María Visitación Hurtado | Miguel
Lastra | Alejandro León | ... | Rubén García);
```

Figura 5. Gramática generada:  
NombreProf.gram

#### 4 Conclusiones y trabajo futuro

En este artículo hemos presentado un sistema de diálogo cuya finalidad es proporcionar información académica en el entorno universitario. Una característica de este sistema es que utiliza gramáticas durante el proceso de RAH cuyo vocabulario no es conocido con antelación a dicho proceso, por estar contenido en una base de datos.

Para resolver este problema de una forma eficiente, hemos desarrollado una técnica que facilita considerablemente la creación de gramáticas estáticas, las cuales son generadas con anterioridad al proceso de RAH, y actualizadas conforme cambian los datos almacenados en la base de datos del sistema. Además, esta técnica es genérica, pues no está ligada a ninguna base de datos en particular.

Una línea de trabajo futuro se centra en añadir al módulo GAG la posibilidad de generar cualquiera de los formatos de gramáticas reconocidos por el W3C para el lenguaje VoiceXML. Es decir, además de JSGF, nos proponemos dar soporte a la generación de gramáticas XML y ABNF. Además, tenemos previsto agregar un gestor de actualizaciones que permita el almacenamiento de plantillas de gramáticas. Cada una de estas plantillas incluirá las gramáticas que se almacenen siempre en el mismo fichero y se generen a partir de los mismos campos de la base de datos. Así, la actualización de dichas gramáticas se podrá realizar de forma transparente con cada cambio efectuado en la base de datos.

#### Bibliografía

Buntschuh, B., Kamm, C., Di Fabrizio, G., Abella, A., Mohri, M., Narayanan, S., Zeljkovic, I., Sharp, R. D., Wright, J., Marcus, S., Shaffer, J., Duncan, R., Wilpon, J. G. 1998. VPQ: A spoken language interface to large scale directory information. *Proc. International Conference on Spoken Language Processing (ICSLP)*. Sidney, Australia, pág. 2863-2867.

D’Haro, L.F., Córdoba, R., Ibarz, I., San-Segundo, R., Montero, J.M., Macías-Guarasa, J., Ferreiros, J., Pardo, J.M. 2004. Plataforma de generación

- semiautomática de sistemas de diálogo multimodales y multilingües: proyecto GEMINI. *Procesamiento del Lenguaje Natural*, 33: 119-126.
- Glass, J., Seneff, S. 2003. Flexible and Personalizable Mixed-Initiative Dialogue Systems. *Proc. Human Language Technology Conference*, Edmonton, Canadá.
- Hartl, A. 2003. A Widget-Based Approach for Creating Voice Applications. *Proc. Physical Interaction*.
- Huuskonen, P., Pavel, D., Tuomela, U. 2003. Context-awareness: A new dimension for communication". *Nokia Research Center*.
- Llisterri, J. 2004. *Las tecnologías del habla para el español*. Sequera, R. (Ed.) Ciencia, tecnología y lengua española: la terminología científica en español. Madrid: Fundación Española para la ciencia y la Tecnología, pág. 123-141
- López-Cózar, R., Rubio, A. J. 1997. SAPLEN: Un Sistema de Diálogo en Lenguaje Natural para una Aplicación Comercial. *Procesamiento del Lenguaje Natural*, 20:65-81.
- López-Cózar, R., Rubio, A.J., Benítez, M.C., Milone, D.H. 2000. Restricciones de Funcionamiento en Tiempo Real de un Sistema Automático de Diálogo. *Procesamiento del Lenguaje Natural*, 26:169-174
- Mc Tear, M. F. 2004. *Spoken Dialogue Technology: Toward the Conversational User Interface*. Springer.
- Nöth, E., Horndasch, A., Gallwitz, F., Hass, J. 2004. Experiences with commercial telephone-based dialogue systems. *Information Technology* 46(6): 315-321
- Rabiner L. R., Juang B. H. 1993. *Fundamentals of Speech Recognition*. Prentice-Hall.
- Schalkwyck, J., Hetherington, L., Story, E. 2003. Speech recognition with Dynamic Grammars Using Finite-State Transducer. *Proc. Eurospeech*, páginas 1969-1972. Ginebra, Suiza.
- Ström, N., Seneff, S. 2000. Intelligent barge-in in conversational systems. *Proc. International Conference on Spoken Language Processing (ICSLP)*, páginas 652-655. Beijing, China.
- Traum, D. R., Larsson, S. 2001. The information state approach to dialogue management. *European Summer School on Logic, Language and Information*.
- Truillet, P., Grisvard, O., Goujon, B. 2004. *SCOPE – CARE II Innovative WP3 – R3 – Model of English*. European Organisation for the Safety of Air Navigation.
- Varela, E. 2004. *Accesibilidad a Internet*. Foro de la Complutense en colaboración con ONCE.