

OPTIMIZACIÓN DEL PROCESO DE GENERACIÓN DE GRAMÁTICAS PARA RAH BASADAS EN REGLAS Y APLICACIÓN EN UN SISTEMA DE DIÁLOGO

Zoraida Callejas Carrión, Ramón López-Cózar Delgado

Dpto. Lenguajes y Sistemas Informáticos, ETS Ingenierías Informática y Telecomunicaciones,
Universidad de Granada, {zoraida, rlopezc}@ugr.es

RESUMEN

En este artículo presentamos una técnica para la actualización de gramáticas para RAH basadas en reglas cuyo vocabulario esté almacenado en una o varias bases de datos. La técnica, denominada CGBD (Creación de Gramáticas Basada en Disparadores), tiene como finalidad actualizar automáticamente el vocabulario de dichas gramáticas mediante un mecanismo de disparadores (triggers) de bases de datos, conforme se producen cambios en el contenido de éstas. Describimos asimismo cómo esta técnica ha sido implementada mediante la herramienta de Generación Automática de Gramáticas (GAG). Los resultados de evaluación, obtenidos usando un sistema de diálogo experimental, muestran que la técnica mejora significativamente la percepción que tienen los usuarios de prueba respecto a la velocidad de interacción del sistema. En comparación con otras técnicas existentes, la técnica CGBD ha sido preferida en la mayoría de las ocasiones, especialmente a partir de un valor umbral de 300.000 palabras en el vocabulario.

1. INTRODUCCIÓN

Las gramáticas para reconocimiento automático del habla (RAH) permiten alcanzar una gran precisión en aplicaciones con un vocabulario y tipo de construcciones gramaticales muy bien delimitados. Es el caso de los sistemas de diálogo diseñados para proveer información, que normalmente tienen en cuenta vocabulario concerniente a las distintas consultas posibles (p.e. horarios de trenes [1], de aviones [2], o información en exposiciones y museos [3]).

Sin embargo, en algunas aplicaciones el vocabulario aceptado por las gramáticas puede ser desconocido a priori por estar almacenado en una (o varias) bases de datos (p.e. nombres de profesores de una universidad) en contraposición con aquellas en que el vocabulario es conocido desde el momento de diseño del sistema (p.e. confirmaciones y negaciones). En la literatura podemos encontrar diversas aproximaciones a la construcción de gramáticas para RAH basadas en reglas, que podemos clasificar en dos grandes grupos: el primero crea las reglas de forma estática, mientras que el segundo lo hace dinámicamente. Ambos enfoques

poseen ventajas y desventajas. Un problema del enfoque estático es que las reglas no se actualizan cuando se producen cambios en las bases de datos y por tanto las frases aceptadas pueden quedarse obsoletas. En el caso de gramáticas dinámicas, una desventaja es que la extracción del vocabulario y la creación de las reglas puede traer consigo tiempos de respuesta del sistema muy grandes. Para solventar estos problemas, presentamos una técnica que crea las reglas estáticamente con vocabulario desconocido a priori, y las actualiza empleando un mecanismo de disparadores cuando se producen cambios en las bases de datos. La versión preliminar de esta técnica fue presentada en [4], en el presente artículo describimos la versión completa y los resultados de su evaluación con usuarios reales.

El artículo está organizado de la siguiente forma. En la sección 2 presentamos el estado del arte de la creación de gramáticas basadas en reglas. En la sección 3 presentamos nuestro enfoque para llevar a cabo esta tarea y describimos la implementación empleada para probar su efectividad. En la sección 4 describimos el sistema de diálogo oral experimental en el que hemos evaluado la técnica y los resultados obtenidos. Finalmente, la sección 5 presenta las conclusiones y propone algunas posibilidades de trabajo futuro.

2. TÉCNICAS EXISTENTES PARA LA CREACIÓN DE GRAMÁTICAS BASADAS EN REGLAS

2.1. Creación estática

La forma más sencilla de construir gramáticas para el RAH basadas en reglas consiste en crear dichas reglas de forma estática. Además, es la forma más rápida si el vocabulario permanece inalterado. De hecho, la mayoría de los entornos comerciales para el desarrollo de sistemas de diálogo oral proveen de herramientas para la creación y prueba de gramáticas basadas en reglas (p.e. IBM WebSphere, Voxeo o Nuance). Sin embargo, cuando el vocabulario no es conocido a priori sino que se encuentra almacenado en una o varias bases de datos, la creación estática puede provocar inconsistencias entre los contenidos de ésta y el vocabulario incluido en las gramáticas.

2.2. Creación dinámica

En la literatura encontramos diversas técnicas para crear gramáticas para RAH dinámicamente cuando el vocabulario aceptado por la aplicación es desconocido a priori. Una primera técnica construye las gramáticas dinámicamente en tiempo de procesamiento [5]. Esta técnica ofrece flexibilidad puesto que las gramáticas generadas de esta manera siempre están actualizadas con los últimos cambios en los datos. Sin embargo, este método puede implicar una carga computacional enorme, que se traduce en un incremento del tiempo de ejecución. En aplicaciones con bases de datos muy grandes, este tiempo puede ser excesivo, provocando que el sistema sea considerado por los usuarios “lento” en su tiempo de respuesta.

Una segunda técnica crea las gramáticas antes de que tenga lugar el proceso de reconocimiento automático del habla. Estas gramáticas se actualizan siempre antes de la ejecución del sistema independientemente de los cambios en las bases de datos [6]. Este método no implica un incremento en el tiempo de reconocimiento, con lo que el incremento en el tiempo de espera es menor desde el punto de vista del usuario, pero sigue habiendo un retardo al comienzo de la ejecución que se ve incrementado al hacerlo el tamaño de las bases de datos.

En ambos casos, la aplicabilidad de la técnica depende del tamaño del vocabulario. Como se ha discutido en diversos trabajos previos (p.e. [7]) hay tres tiempos de ejecución básicos a ser considerados:

- El límite para que le usuario considere que la aplicación se realiza en tiempo real es de 0.1 segundos.
- De 0.1 a 1.0 segundos el usuario nota un retardo pero su flujo de pensamiento no se interrumpe.
- De 1.0 a 10.0 segundos, el usuario todavía presta atención al diálogo pero para retardos mayores es necesario darle alguna información con un mensaje del sistema y/o una melodía para indicarle que el sistema se encuentra procesando la respuesta [8].

Hemos medido en nuestro laboratorio el tiempo de ejecución necesario para construir una gramática de una única regla de gramática a partir de una columna específica de una base de datos, teniendo en cuenta el número de palabras existentes en la misma. Como se muestra en la Figura 1, el límite 0.1 se alcanza con 10.000 palabras, el límite de 1 segundo con 100.000 palabras, y el límite máximo (10 segundos) se alcanza con vocabularios entre 300.000 y 400.000 palabras.

Por ejemplo, para un vocabulario de un millón de palabras obtenemos un retardo de 30 segundos. Por tanto, para vocabularios grandes (más de 300.000 palabras) el usuario se percata de la sobrecarga introducida y se necesita una técnica de

retroalimentación que le mantenga informado de los tiempos de espera. Desafortunadamente, estas técnicas suelen causar una mala impresión a los usuarios, que normalmente las encuentran muy irritantes [9]. Además, un retardo en la respuesta del sistema lo más pequeño posible, definido éste como el tiempo transcurrido entre que el usuario termina de hablar y el sistema responde [10], es vital para minimizar los costes de interacción. Esto, unido con la maximización del éxito en la tarea, es vital para mejorar el grado de satisfacción de los usuarios y se emplea como parámetro clave en la evaluación de sistemas de diálogo, tal y como se propone en el marco PARADISE [10].

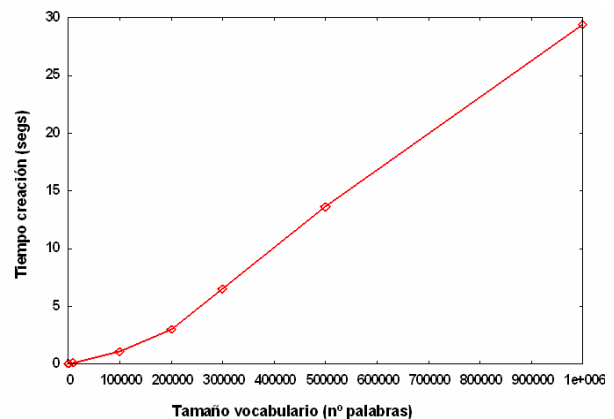


Figura 1. Latencia en el proceso de creación dinámica de gramáticas para RAH basadas en reglas.

3. TÉCNICA CGBD

Para evitar los problemas comentados en la sección anterior, proponemos una nueva técnica denominada CGBD (Creación de Gramáticas Basada en Disparadores) para optimizar el proceso de creación de gramáticas para RAH basadas en reglas. La idea de la misma es actualizar el vocabulario de dichas de forma automática conforme se producen cambios en las bases de datos utilizadas, empleando para ello un mecanismo de disparadores. De esta forma, no hay retardo de creación en tiempo de ejecución y el usuario siente que está interactuando con un sistema rápido.

La mayor ventaja de esta técnica en comparación con otras que encontramos en la literatura (p.e. [11]) es que es general y por tanto adecuada para crear gramáticas que puedan emplearse en cualquier sistema de diálogo que extraiga vocabulario de bases de datos.

El proceso de creación de dichas gramáticas se lleva a cabo en tres pasos:

- Extracción de información de las bases de datos. Las gramáticas se generan antes de que el proceso de RAH se lleve a cabo.
- Construcción de las gramaticales a partir de los datos extraídos. Los datos se disponen en

- formato JSGF (Java Speech Grammar Format) o ABNF (Augmented Backus-Naur Form).
- iii) Actualización de las gramáticas con los cambios efectuados en las bases de datos. Para asegurar que las gramáticas están actualizadas permanentemente incluso cuando cambia el contenido de las bases de datos, nuestra técnica emplea un mecanismo basado en “disparadores” (triggers) de bases de datos. Éstos se disparan cuando se producen cambios en los campos de las bases de datos a partir de los cuales se extraen las palabras para las reglas de las gramáticas.

El grupo de trabajo W3C Voice Browser especifica en [12] que las reglas debe ser re-definibles en tiempo de ejecución. Para ello, apuntan diversos mecanismos. Entre ellos, subrayamos la división del espacio de reglas en estáticas y dinámicas. Empleando la técnica CGBD, la estructura de frase se crea estáticamente conteniendo referencias a reglas dinámicas, que pueden almacenarse en el mismo fichero o en uno externo, como ilustra el siguiente ejemplo:

```
#ABNF 1.0 iso-8859-1;
root $consulta;
public $consulta = Me gustaría hablar con
$profesor;
public $profesor = [Don | Doña] $<Path de la
regla para el nombre> $<Path de la regla para el
primer apellido> $<Path de la regla para el
Segundo apellido >;
```

Empleando esta gramática, los usuarios pueden expresar su intención de hablar con un profesor diciendo su nombre y apellidos. Las reglas \$nombre y \$apellidos se construirían con el vocabulario extraído de una base de datos en la que se almacenasen los nombres de profesores. Por tanto, estas reglas pueden incluirse en el mismo archivo donde se encuentra toda la gramática o crearse en ficheros independientes como en el ejemplo anterior.

3.1. Implementación

La herramienta de Generación Automática de Gramáticas (GAG), que desarrollamos en un trabajo previo ([4]), realiza la creación de gramáticas de forma automática. Para este trabajo, le hemos añadido la funcionalidad de la técnica CGBD empleando PHP, HTML, JavaScript y PostGreSQL. La herramienta se emplea como un módulo adicional del sistema de diálogo oral UAH (que describimos en la sección 4.1). La herramienta dispone de una interfaz sencilla de utilizar para permitir al diseñador del sistema escoger los campos de la base de datos que se emplearán para extraer el vocabulario. El proceso para crear reglas empleando esta herramienta se divide en tres etapas:

- En primer lugar, la herramienta GAG solicita al usuario el nombre de la base de datos, el nombre del ordenador servidor, el nombre de usuario y la contraseña para acceder a la base de datos donde se encuentra almacenada la información. También solicita el nombre de la regla de gramática y el tipo de gramática a crear (JSGF o ABNF).
- En segundo lugar, se visualizan los campos de las tablas de la base de datos en un menú *drop-down*. Los campos aparecen en el menú en el mismo orden en que se encuentran en la tabla correspondiente, pero el diseñador puede seleccionarlos en cualquier orden para crear la regla de la gramática. Cuando se selecciona un campo, un número aparece automáticamente junto al campo indicando el orden de selección. Por ejemplo, en una aplicación académica el diseñador del sistema podría seleccionar el nombre de profesor y seguidamente su apellido obteniendo la regla: <profesor> = (“Sergio García”|...| “Miguel Moreno”). Alternativamente, podría seleccionar ambos casos en orden inverso, es decir, primero el apellido y después el nombre, en cuyo caso la regla obtenida sería: <profesor> = “García Sergio”|...| “Moreno Miguel”).
- Por último, después de la selección de campos el diseñador debe introducir el nombre del archivo donde desea almacenar la regla de gramática. Puede elegir, además, entre crear la regla como parte de una gramática existente en su mismo fichero, o bien, crear un único fichero con una nueva gramática cuya única regla sea esa.

La herramienta GAG implementa el mecanismo de disparadores CGBD para mantener las reglas de las gramáticas actualizadas con los últimos cambios realizados en las bases de datos. Para ello, al final de la creación de cada gramática el diseñador del sistema indica si desea que se actualice automáticamente el vocabulario en la regla con cambios en la base de datos. En caso afirmativo, se crean dinámicamente disparadores que se activan cuando los valores correspondientes se actualizan, borran o insertan. Por ejemplo, si se crea una regla para los nombres y apellidos de profesores, el disparador incluirá “María” como una nueva palabra en la regla si se introduce en el campo “Nombre” de la tabla “Profesor”. También borra la palabra “María” de la regla si se borra de la tabla, y cambia “María” por “Mónica” si el campo se actualiza con este nuevo valor.

Como los disparadores indican los valores antiguos y nuevos del campo de la tabla, sólo se realizan los cambios pertinentes en las reglas de las diversas gramáticas en lugar de generarlas por completo de nuevo cada vez que ocurra cualquier cambio, lo que incluiría un retardo equivalente al de crearlas

dinámicamente (como se ha comentado en la sección 2.2). El proceso completo de actualización automática se ilustra en la Figura 2, donde se esquematiza una arquitectura distribuida en la que el RAH se realiza en un ordenador con que dispone de una tarjeta de interfaz telefónica Intel Dialogic D/41JCT-LS, mientras que el gestor de diálogo y la herramienta GAG se ejecutan en el ordenador servidor, mientras que las bases de datos se almacenan en un equipo diferente.

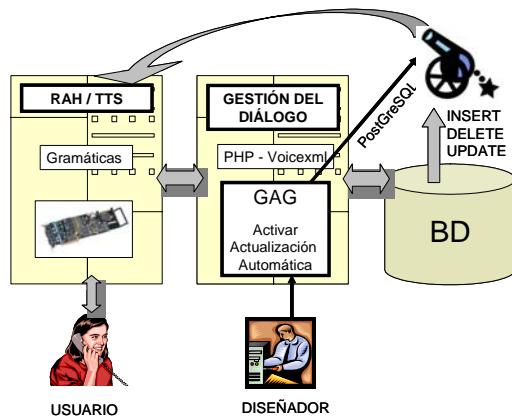


Figura 2. Actualización automática de gramáticas con la herramienta GAG

4. EXPERIMENTOS

Para obtener resultados experimentales usando la técnica CGBD hemos empleado el sistema de diálogo oral UAH (Universidad Al Habla) [13]. En esta sección describimos brevemente el sistema y a continuación discutimos los resultados obtenidos en la evaluación. Realizamos dos tipos de experimentos. En primer lugar, determinamos el grado de satisfacción de los usuarios conforme varía el tamaño del vocabulario, teniendo en cuenta únicamente el retardo introducido por el proceso de creación de las gramáticas. En segundo lugar, observamos el efecto producido por la técnica en la interacción de usuarios reales con el sistema UAH. Nuestro objetivo, en este caso, es conocer el grado de aceptación del funcionamiento del sistema, de una forma global.

4.1. El sistema de diálogo UAH (Universidad al Habla)

El sistema UAH ha sido desarrollado en nuestro laboratorio para proporcionar acceso telefónico a la información disponible en la página web de nuestro departamento, permitiendo de esta manera el acceso a esta información a usuarios invidentes. El sistema se encuentra en funcionamiento desde Julio de 2005 en nuestro laboratorio prestando un servicio de información telefónica acerca de nuestro Departamento y asimismo, de forma limitada, acerca del proceso de auto-matrícula en nuestra Universidad.

En el momento de la escritura de este artículo el sistema está en proceso de evaluación. Para ello grabamos las llamadas de los usuarios, clasificamos las frases pronunciadas teniendo en cuenta los *prompts* del sistema, y almacenamos en ficheros las hipótesis de reconocimiento obtenidas (frases reconocidas). Además, obtenemos las opiniones de los usuarios acerca de distintos aspectos de la interacción con el sistema a través de formularios web¹. De esta forma, hemos construido un corpus de pronunciaciones de frases que podemos analizar para obtener medidas objetivas del funcionamiento del sistema. Además, gracias a los formularios web podemos obtener también medidas subjetivas de evaluación del sistema [14].

Las gramáticas para RAH que usa el sistema se crean de formas diversas dependiendo del tipo de vocabulario y del momento de creación del mismo. En total empleamos cuatro métodos de creación, atendiendo a que el vocabulario sea conocido o desconocido en tiempo de diseño. Es decir, consideramos por una parte vocabulario que no cambia nunca, y por otra, vocabulario que puede ser cambiante por ser actualizado en una o varias bases de datos. En función del tipo de vocabulario realizamos la creación de las reglas de las gramáticas de forma estática o dinámica. La creación dinámica es preferible cuando el vocabulario de la gramática es conocido a priori, o en caso de ser desconocido, es lo suficientemente pequeño como para permitir la creación en tiempo de ejecución sin introducir un gran retardo. Empleando la técnica propuesta en este artículo, si el vocabulario es muy grande, usamos la herramienta GAG para crear automáticamente las reglas de las gramáticas. Para ello, la estructura de las frases se almacena en una gramática general que hace referencia a reglas externas que contienen el código generado dinámicamente (como se describe en la sección 3.1).

4.2. Resultados

En los experimentos realizados comparamos dos enfoques distintos para la creación de gramáticas para RAH basadas en reglas a partir de datos extraídos de bases de datos: creación dinámica (método tradicional) y creación usando la técnica propuesta (CGBD).

Para realizar una comparativa lo más ilustrativa posible de ambas técnicas, empleamos vocabularios que contenían desde 1 a 10⁶ palabras (como los mostrados en la Figura 1). Sin embargo, el tamaño del vocabulario manejado en UAH es inferior a las 4.000 palabras, con lo que en muchas ocasiones se tuvieron que emplear vocablos de relleno generados automáticamente en la base de datos desde la cual se extrae el vocabulario para generar las reglas de las gramáticas de RAH.

Con la configuración anterior, pedimos a 30 usuarios de prueba que emplearan el sistema dos veces.

¹ El lector puede participar en la evaluación del sistema accediendo a la siguiente URL: <http://teide.ugr.es/UAH/InstruccionesUAH.html>

En primer lugar con la técnica de generación dinámica, pidiendo que expresaran su grado de satisfacción (en una escala de 1 a 5 puntos) acerca de la velocidad de interacción con el sistema; y en segundo lugar, empleando la técnica CGBD, que eligieran entre ésta y la primera estrategia utilizada. Los resultados de los experimentos anteriores son los mostrados en la Figura 3, donde se aprecia que el grado de satisfacción de los usuarios respecto a la creación dinámica de las gramáticas decae conforme aumenta el tamaño del vocabulario. En cambio, la tendencia es opuesta cuando se usa la técnica propuesta en este artículo.

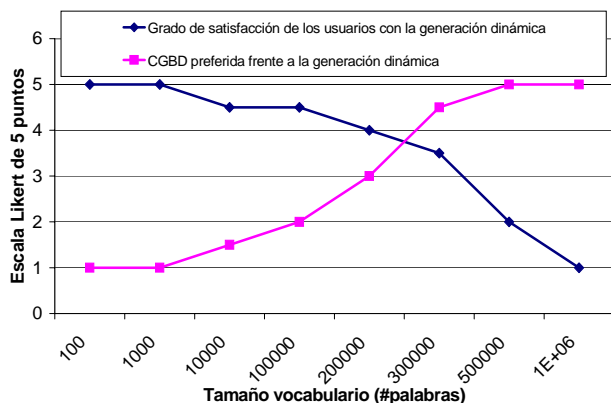


Figura 3. Valoración de técnica la técnica CGBD frente a la creación dinámica de gramáticas

Es claramente destacable la correspondencia entre el grado de satisfacción de los usuarios y las restricciones de tiempo comentadas en la sección 2.2. Como podemos observar en la Figura 3, para vocabularios menores de 10.000 palabras (que conllevan un tiempo de generación con la herramienta GAG menor de 0.1 segundos) los usuarios no son conscientes del retardo del sistema cuando se usa la técnica de generación dinámica. También podemos observar que existe un punto en el que ambas líneas se cruzan: cuando el vocabulario tiene un tamaño de unas 300.000 palabras. La figura muestra que a partir de este punto, el tamaño del vocabulario es lo suficientemente grande como para hacer que la técnica propuesta sea claramente preferible a la creación dinámica, dado el retardo causado por ésta última.

En general, al emplear el módulo GAG en el sistema UAH tenemos que, independientemente del tamaño del vocabulario, el grado de satisfacción de los usuarios con la interacción (medida en una escala de Likert de 5 puntos) es en el 78% de los casos, mayor o igual a 3 (como muestra la Figura 4).

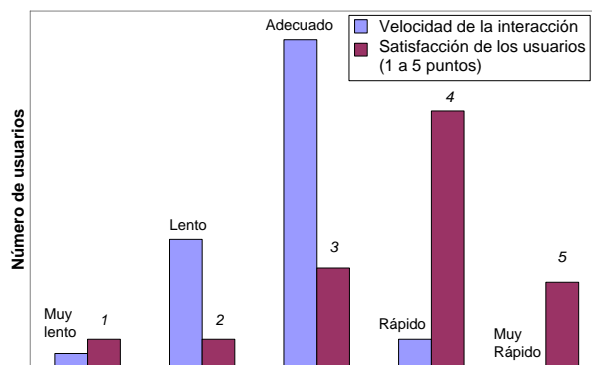


Figura 4. Medidas subjetivas acerca de la velocidad de interacción y el grado de satisfacción de los usuarios.

5. CONCLUSIONES Y TRABAJO FUTURO

Algunos sistemas de diálogo oral emplean gramáticas para RAH basadas en reglas cuyo vocabulario se extrae de bases de datos. Este método causa problemas si el tamaño de las bases de datos es grande, puesto que la interacción se ralentiza debido al tiempo requerido para crear las gramáticas en tiempo de ejecución. Para evitar este problema, otros sistemas de diálogo crean las reglas de las gramáticas de forma estática. Sin embargo, usando este método pueden surgir inconsistencias entre el vocabulario de las gramáticas y el contenido de las bases de datos conforme dicho contenido cambia.

Para resolver este problema de forma eficiente, hemos desarrollado una técnica que crea las reglas de las gramáticas antes de que comience el proceso de RAH, de forma tal que el vocabulario de dichas reglas se actualiza automáticamente conforme el contenido de las bases de datos cambia. Esta técnica es genérica y no está ligada a ningún tipo de base de datos o dominio de aplicación en particular. Por tanto, puede aplicarse para mejorar la velocidad de interacción de cualquier sistema de diálogo que utilice gramáticas para RAH basadas en reglas cuyo vocabulario deba ser extraído de bases de datos.

La técnica ha sido aplicada a un sistema de diálogo oral que proporciona información acerca de nuestro Departamento y, de forma limitada, acerca del proceso de auto-matrícula en nuestra Universidad. Los resultados experimentales muestran que usando dicha técnica el grado de satisfacción de los usuarios que han participado en la evaluación es alto pues éstos consideran que la interacción se realiza a una velocidad adecuada.

Como trabajo futuro tenemos previsto realizar estudios más profundos de la apreciación de los usuarios respecto de la velocidad de interacción del sistema usando las dos técnicas de creación de gramáticas estudiadas: esto es, técnica tradicional (creación dinámica de gramáticas) frente a técnica CGBD. Nuestro interés se centrará, fundamentalmente, en determinar el efecto producido por el uso de una u

otra técnica junto con diversas técnicas de gestión de la iniciativa en el diálogo (esto es, iniciativa dirigida por sistema, dirigida por el usuario o mixta).

6. BIBLIOGRAFÍA

[1] D. Griol, F. Torres, L.F. Hurtado, E. Sanchis, E. Segarra, "Different approaches to the dialogue management in the DIHANA project", SPECOM 2005

[2] S. Seneff, J. Polifroni, "Dialogue management in the mercury flight reservation system", Proc. ANLP-NAACL Sat. Workshop, Seattle, 2000.

[3] A. Ndiaye, P. Gebhard, M. Kipp, M. Klesen, M. Schneider, W. Wahlster, "Ambient Intelligence in Edutainment: Tangible Interaction with Life-Like Exhibit Guides", Artificial Intelligence, Vol. 3814, pp. 104-113, 2005.

[4] Z. Callejas, R. López-Cózar, "Nueva Técnica de Generación Automática de Gramáticas para Sistemas de Diálogo", Procesamiento del lenguaje natural, vol 35, pp. 205 - 212

[5] P. Truillet, O. Grisvard, B. Goujon, "SCOPE – CARE II Innovative WP3 –R3 – Model of English", European Organisation for the Safety of Air Navigation, 2004.

[6] J. Schalkwyck, L. Hetherington, E. Story, "Speech recognition with Dynamic Grammars Using Finite-State Transducer", Proc. Eurospeech, 1969-1972. Geneva, Switzerland, 2003.

[7] J. Nielsen, "Usability Engineering", Ed. Morgan Kaufmann, San Francisco, 1994.

[8] L. Cerrato, "A comparison between feedback strategies in human-to-human and human-machine communication", Proc. ICSLP, 557-560. Denver, USA, 2002.

[9] K. Mäkelä, E. Salonen, M. Turunen, J. Hakulinen, R. Raisamo, "Evaluating the User Interface of a Ubiquitous Computing system Doorman", Proc UbiComp, Atlanta, USA, 2001.

[10] S. Möller, "Quality of Telephone-Based Spoken Dialogue Systems", Ed. Springer, New York, 2005.

[11] M.F. Mc Tear, "Spoken Dialogue Technology: Toward the Conversational User Interface", Ed. Springer, 2004.

[12] W3C Working Draft 23 Dec 1999, Grammar Representation Requirements for Voice Markup Languages. Available in <http://www.w3.org/TR/voice-grammar-reqs/>

[13] Z. Callejas, R. López-Cózar, "Implementing Modular Dialogue Systems: A Case of Study", Proc. ASIDE, Aalborg, Denmark, 2005.

[14] R. López-Cózar, M. Araki, "Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment", Ed. Wiley, England, 2005.