Un sistema de diálogo multilingüe dirigido por la semántica

Marta Gatius
TALP Research Center
Software Department
Universitat Politècnica de Catalunya
Jordi Girona 1-3,08034 Barcelona
gatius@lsi.upc.edu

Meritxell González
TALP Research Center
Software Department
Universitat Politècnica de Catalunya
Jordi Girona 1-3,08034 Barcelona
mgonzalez@lsi.upc.edu

Resumen: Este artículo presenta un sistema de diálogos multilingüe basado en la semántica. El sistema utiliza una ontología que modela la aplicación para gestionar de forma eficiente la interacción oral y textual en diferentes lenguas (inglés, castellano y catalán). El conocimiento de la aplicación es utilizado por el gestor de diálogo para determinar la estructura del diálogo. También se utiliza para generar las gramáticas y léxicos en las diferentes lenguas. Estos recursos lingüísticos incorporan información de la aplicación para facilitar la interpretación semántica de las intervenciones del usuario.
Palabras clave: Sistemas de diálogo, ontologías, procesamiento semántico, interacción multilingüe.

Abstract: This article presents a semantic-driven multilingual dialogue system. The system described supports both speech and textual interaction. The system uses ontologies representing the application knowledge to support a friendly communication in several languages: English, Spanish and Catalan. Application information is incorporated into the grammars and lexicon in order to simplify the semantic processing. The application model is also used by the Dialogue Manager.
Keywords: Dialogue Systems, ontologies, semantic processing, multilinguality.

1 Introducción

Los recientes avances en las tecnologías del habla, así como la creciente demanda de sistemas de iteración amigables sobre diferentes canales (teléfono, web, agendas digitales personales,...) ha espoleado el desarrollo de sistemas de diálogo de gran utilidad en la vida cotidiana.

En este trabajo se presenta el primer prototipo de diálogo desarrollado en el proyecto europeo HOPS (*Enabling an Intelligent Natural Language Based Hub for the Deployment of Advanced Semantically Enriched Multi-channel Mass-scale Online Public Services*), descrito en (2). El objetivo principal del proyecto HOPS es permitir el acceso mediante la voz a distintos servicios de las administraciones que participan en el proyecto (el Ayuntamiento de Barcelona, el Ayuntamiento de Torino y el Borough of Camden, en Londres). Como una extensión natural, la plataforma a desarrollar también soportará diálogos de entrada textual (a través de la web o de otros canales). Además, la plataforma integrará tecnologías de la Semantic Web.

Este artículo se centra en la gestión del diálogo y en el procesamiento de la entrada textual en el prototipo desarrollado para un servicio en particular: el de recogida de muebles. Este sistema soporta diálogos telefónicos y diálogos de texto (a través de la web) en varias lenguas: inglés, castellano y catalán. El sistema utiliza la representación del conocimiento de la aplicación en las diferentes fases del proceso comunicativo, es decir, en el control del diálogo y en el procesamiento de la entrada textual y oral.

Desde hace años se ha venido utilizando la semántica de la aplicación en el procesamiento y en la generación del lenguaje natural, (Mahesh and Nirenburg, 1995), (Bateman,

Magnini and Rinaldi, 1994). La representación independiente del conocimiento de la aplicación se ha mostrado especialmente útil en sistemas que soportan más de una lengua, ya que permite representar de forma explícita la relación entre un concepto de la aplicación y su expresión en diferentes lenguas.

En los sistemas de diálogo, la utilización de una representación del conocimiento de la aplicación y del control de la interacción en bases independientes es cada vez más frecuente. No sólo supone mejorar la calidad del diálogo (Milward, 2004), sino que además permite que el sistema se pueda adaptar fácilmente a diferentes aplicaciones (D'Haro, 2004), (Quesada, 2002) , (Rodrigo, García, Martínez, 2002).

La Sección 2 presenta una visión general del sistema y sus tres componentes: el módulo de voz, el módulo de texto y el gestor del diálogo. En las siguientes secciones se describe la utilización de la información de la aplicación en cada uno de estos tres componentes. En la última sección se plantean las conclusiones y el trabajo futuro.

2 Descripción general

La función principal del sistema desarrollado consiste, como ya se ha comentado en la introducción, en mantener una comunicación amigable (a través del teléfono o la web) que guíe al usuario del servicio de recogida de muebles que gestiona el Ayuntamiento de Barcelona.

2.1 Arquitectura del sistema

El sistema sigue un diseño modular, como puede observarse en la Figura 1. Los módulos del sistema son los siguientes:

- El módulo de voz
- El módulo de texto
- El gestor de diálogo

La conexión entre los diferentes componentes de la plataforma (el gestor de diálogo, los componentes que intervienen en el procesado de voz y texto, el back-end de la aplicación) se realiza a través de FADA, una herramienta diseñada para facilitar la interoperabilidad entre procesos heterogéneos, descrita en (1).

2.1.1 El gestor de diálogo

El gestor de diálogo recibe del módulo de voz y del módulo de texto la información obtenida del usuario. Tanto si se trata de una interacción a través del texto o de la web, el gestor de diálogo recibe la interpretación semántica obtenida al procesar la intervención del usuario. A partir de esta información el gestor analiza el nuevo estado del diálogo y determina la siguiente acción a realizar, que puede ser una llamada al back-end de la aplicación o una nueva interacción con el usuario. Para determinar la siguiente acción, el gestor de diálogo utiliza información sobre la aplicación, como se explicará más adelante.

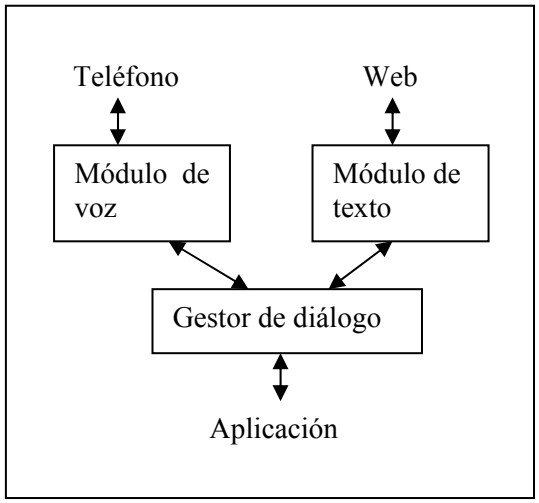

Figura1: Arquitectura del sistema

2.1.2 El módulo de voz

El módulo de voz se basa en los componentes de la plataforma VoiceXML desarrollada por Loquendo (3). Este módulo es el encargado de gestionar la interacción telefónica con el usuario y procesar su intervención. La interpretación semántica resultante se pasa al gestor de diálogo. El gestor de diálogo devolverá al módulo de voz la dirección en la que se encuentra el fichero VoiceXML que se debe utilizar en la siguiente interacción con el usuario. Este fichero VoiceXML puede haber sido creado por el diseñador antes de iniciar la comunicación o generarse dinámicamente.

Al iniciar la comunicación el módulo de voz preguntará al usuario la lengua en la que desea expresarse y pasará esta información, junto con el identificador de usuario, al gestor de diálogo.

2.1.3 El módulo de texto

El módulo de texto integra dos componentes: un servidor de texto para gestionar la entrada y la salida a través de la web y un analizador que realiza el procesado sintáctico y semántico de la entrada del usuario, desarrollado por (Gatius, 2001). El módulo de texto pasa la interpretación resultante al gestor de diálogo y éste le retorna el texto que debe aparecer por pantalla en la siguiente interacción.

Al iniciar una conversación con un nuevo usuario, el servidor de texto pregunta al usuario la lengua en que desea expresarse. Esta información es almacenada, juntamente con el identificador del usuario. En cada interacción se pasa al analizador el texto entrado y el identificador que indica la lengua.

2.2 Incorporación del conocimiento de la aplicación

La lógica de la interacción para una aplicación concreta, así como la adaptación de los recursos que soportan la comunicación (gramáticas y léxicos) debe realizarse analizando cuidadosamente los servicios que la aplicación puede realizar (las operaciones) y la información que se necesita del usuario. Por esta razón, es recomendable representar de forma explícita toda la información de la aplicación que puede aparecer durante la comunicación. Esta información debe obtenerse a partir del estudio de las funcionalidades de la aplicación y de las conversaciones que tienen lugar con interlocutores humanos (si es que pueden obtenerse).

En nuestro sistema todos los conceptos de la aplicación que intervienen en la comunicación se representan en una ontología, incorporada al gestor de diálogo. En esta ontología, cada concepto está descrito por un conjunto de atributos cuyo valor se le preguntará o se le comunicará al usuario durante la conversación. Para la aplicación de recogida de objetos se han descrito tres conceptos, correspondientes a los tres servicios que ofrece la aplicación: dar información, concretar una fecha de recogida y cancelar recogida. La Figura 2 muestra el concepto que representa una recogida (**collection**). Este concepto se describe por el conjunto de atributos que representa la información que la aplicación necesita: el tipo de objeto a recoger (**object)**, si es contaminante o no (**pollutant**), el nombre (**name**), la dirección (**address**) y el teléfono (**telephone**) del usuario.

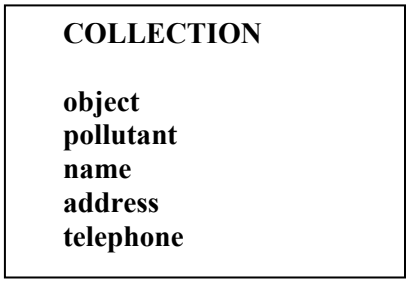

Figura 2: *El concepto que representa el servicio de recogida de un objeto*

Las gramáticas y léxicos para procesar las intervenciones del usuario en las diferentes lenguas, así como el texto que representa las intervenciones del sistema, se obtienen a partir de esta representación. Cada uno de los atributos que describe el concepto se relaciona con los mensajes que deberá presentar el sistema para obtener su valor en las diferentes lenguas. Por ejemplo, el atributo **object**, que determina el tipo de objeto a recoger se asocia, para la interacción en castellano con la pregunta *Qué objeto quiere tirar?* y para la interacción en inglés con la oración *What object do you want to get ridden of?*. El atributo también se puede asociar con las reglas gramaticales necesarias para reconocer la respuesta del usuario a esta pregunta.

3 La utilización de la semántica en el módulo de voz

Los componentes de voz están basados en el formalismo estándar VoiceXML(4). VoiceXML es un lenguaje para diseñar diálogos que incluye especificaciones relativas a los diferentes componentes que intervienen en el reconocimiento y síntesis de voz. Los sistemas VoiceXML utilizan modelos de lenguaje basados en gramáticas. Aunque durante la última década se han realizado distintos trabajos

de investigación en reconocimiento de voz siguiendo modelos de lenguaje estadísticos, la mayoría de sistemas comerciales siguen una aproximación basada en gramáticas. Las motivaciones son tanto de índole práctico como teórico, siendo la principal de ellas que los métodos que utilizan gramáticas, a diferencia de los métodos estadísticos, no necesitan un gran corpus de ejemplos. En los sistemas VoiceXML las gramáticas representando las intervenciones de los usuarios se utilizan tanto en el reconocimiento de la entrada oral como en su procesamiento morfológico, sintáctico y semántico. El gran éxito de los sistemas VoiceXML se debe principalmente a que son fáciles de desarrollar, se puede implementar una aplicación sin conocer los detalles técnicos de los diferentes componentes de voz implicados. Además, se facilita su adaptación a servicios web. Por otra parte, al utilizar un formalismo estándar estos sistemas se pueden adaptar a cualquier plataforma VoixeXML.

En el sistema que presentamos se han utilizado gramáticas basadas en el formalismo Java Speech Grammar Format (JSGF), que permite incorporar información semántica a las reglas. La generación de las gramáticas a partir de la representación formal de la aplicación no sólo nos permite escribir las reglas que recogen las posibles intervenciones del usuario para cada pregunta o mensaje del sistema, sino también asociarles la interpretación semántica más adecuada. Como se ha comentado, cada atributo describiendo a un concepto tiene asociado un mensaje (o pregunta) y unas reglas gramaticales. La interpretación obtenida al procesar la respuesta utilizando esas reglas será el identificador del concepto, el atributo y su valor. Independientemente de la lengua que se utilice, la interpretación semántica siempre será en inglés, la lengua en la que está la representación de la aplicación.

4 La utilización de la semántica en el módulo de texto

El módulo de texto se basa en un analizador que realiza el procesamiento sintáctico y semántico del texto introducido por el usuario. Este analizador realiza la interpretación semántica en paralelo con la sintáctica, siguiendo una de las aproximaciones más simples al procesamiento semántico, el análisis semántico dirigido por la sintaxis. Esta aproximación consiste en obtener las representaciones semánticas a partir solamente del conocimiento del léxico y la gramática. Se basa en el principio de composicionalidad, es decir, el significado de una oración puede obtenerse a partir de sus constituyentes. En esta aproximación se incorpora información semántica a las reglas y a las entradas léxicas.

La gramática y el léxico utilizados para cada lengua se han desarrollado especialmente para la aplicación de recogida de objetos. Se ha decidido utilizar gramáticas y léxicos restringidos a la aplicación por motivos de eficiencia: se reduce la ambigüedad y se simplifica la interpretación semántica. Las gramáticas utilizadas no son tan simples como las incorporadas en el módulo de voz, dado que se permite cierta iniciativa al usuario en sus intervenciones. A continuación se describe la información de la aplicación incorporada a la gramática y el léxico y su procesamiento por el analizador.

4.1 El léxico

El léxico contiene palabras y nombres compuestos como entradas léxicas. Algunas de estas entradas se han obtenido directamente de la representación semántica de la aplicación y corresponden a conceptos, atributos y valores utilizados en el modelado de la aplicación. Otras son generales y, por tanto, utilizables en cualquier aplicación (por ejemplo, las entradas correspondientes al verbo *ser*).

Las entradas léxicas consisten en tres campos: categoría, interpretación semántica y cadena. Las categorías pueden incorporar información semántica (la categoría asociada a un verbo puede ser **v** o, en el caso de que represente un concepto **vconcept**). Las categorías pueden ser aumentadas con rasgos sintácticos y semánticos. Por motivos de claridad, no se han añadido estos rasgos a los ejemplos.

La interpretación semántica asociada con las entradas léxicas se representa en el formalismo *lambda calculus*, una extensión del cálculo de predicados de primer orden que permite trabajar con funciones con parámetros. La interpretación semántica asociada a entradas representando elementos de la aplicación puede consistir en una función lambda o en un valor. Esta información está en inglés, porque los

identificadores utilizados en la representación semántica de la aplicación están en inglés.

La interpretación semántica asociada a entradas generales es siempre una función. La Figura 3 muestra tres ejemplos de entradas léxicas. Las dos primeras que representan información obtenida del modelado de la aplicación. La última representa una entrada general. La primera entrada léxica representa la palabra *tirar*. Su categoría es **vconcept** porque está asociada a un concepto de la ontología que representa la aplicación. Se trata del concepto **collection**. El concepto **collection** está descrito por el atributo **object**. La interpretación semántica asociada a esta entrada es **((l,X),(collection, object, X))**. Se trata de una función lambda con un parámetro (representado por **(l,X**)). Cuando ésta función se aplica sobre un valor (parámetro) en la segunda parte de la función se sustituye la X por el valor. Por ejemplo, si se aplica sobre el valor **table**, el resultado de la interpretación será: **(collection, object, table)**.

cadena	**categoría**	**interpretación**
tirar	vconcept	((l,X),(collection, object, X))
mesa	nvalattribute	(table)
una	det	((l,X)(X))

Figura 3: Ejemplos de entradas léxicas

La segunda entrada de la tabla representa la palabra *mesa*. La categoría es **nvalattibute**, indicando que se trata de un nombre que representa el valor de un atributo. La interpretación semántica es el valor **table**. La tercera entrada representa el determinante indefinido *una*, cuya interpretación semántica es una función con un solo parámetro. Como puede observarse, el hecho de que la interpretación resultante sea sólo el valor sobre el que se aplica la función indica que no aporta información semántica, se trata únicamente de un functor.

4.2 La gramática

La gramática se representa en un formalismo de gramáticas de claúsulas definidas. Se ha utilizado este formalismo porque es más expresivo que el de las gramáticas libres de contexto y porque en un dominio limitado, puede analizarse eficientemente (utilizando Prolog).

La parte izquierda de la regla corresponde a la categoría asociada con una estructura lingüística y la parte derecha de la regla es una secuencia de categorías representado los constituyentes. Por ejemplo, a continuación se muestran las reglas gramaticales utilizadas para representar la expresión en castellano de la oración "*Quiero tirar una mesa*":

(1) s-> vquerer gvconcept (1 2)
(2) gvconcept -> vconcept det nvalattribute (1(2 3))

La parte izquierda de la primera regla es la categoría *s*, representando la oración y la parte derecha representa sus constituyentes: **vquerer** (el verbo) y **gvconcept** (el grupo verbal correspondiente a un concepto que representa el objeto directo del verbo).

La segunda regla representa el grupo verbal "*tirar una mesa*" que expresa el concepto recogida (**collection**). Las categorías que aparecen a la derecha de la regla son: **vconcept** (corresponde a *recogida*), **det** (*una*) y **nvalattribute** (*mesa*).

La información semántica asociada a cada regla indica el orden de interpretación de sus constituyentes. Esta información consiste en una lista de los números correspondientes a los constituyentes y se representa recursivamente como una lista de dos elementos. Cada elemento en la lista puede ser un número o una lista de dos elementos, cada uno de los cuales puede ser, a su vez, un número o una lista.

En tiempo de ejecución, cuando el analizador ha reconocido todos los constituyentes de la regla, los interpreta siguiendo la lista asociada a las reglas. El resultado de evaluar el primer elemento de la lista se aplica al resultado de evaluar el segundo elemento. Si el elemento es un número, el resultado de su evaluación es la interpretación semántica asociada al constituyente representado por el número. Si el elemento es una lista, el resultado se obtiene al aplicar el resultado de evaluar el primer elemento sobre el resultado de evaluar el segundo.

Por ejemplo, la interpretación semántica asociada a la segunda regla es (1 (2 3)), indicando que la interpretación del primer constituyente se ha de aplicar sobre la del segundo. Si utilizamos las entradas definidas en la Figura 3, el primer constituyente corresponde a la categoría **vconcept** y tiene como interpretación (**((l,X),(collection, object, X))**). El segundo constituyente es una lista y por tanto se debe evaluar primero el resultado de aplicar el primer elemento de la lista sobre el segundo. Utilizando las entradas de la Figura 3 el resultado obtenido sería **table**, de manera que la función (**((l,X),(collection, object, X))**) se aplicaría sobre el valor **table** y el resultado sería: **(collection, object, table)**.

Las gramáticas utilizadas en el módulo de texto contienen reglas que permiten reconocer la respuesta del usuario a las preguntas del sistema sobre los valores de atributos concretos, así como reglas que permiten reconocer intervenciones del usuario más complejas, como *Vivo en la calle Diagonal y quiero tirar un electrodoméstico, cuando me la recogerán?*. En este ejemplo la interpretación semántica incluye el valor de dos atributos del concepto *recogida*.

4.3 El analizador

El analizador que hemos utilizado es un analizador de *charts* basado en la unificación de Prolog. La implementación se basa en una versión modificada del algoritmo *left-corner* descrito en (Ross, 1982). Las modificaciones sobre dicho algoritmo se han realizado para adaptarlo a gramáticas de cláusulas definidas en las que las categorías pueden estar aumentadas con rasgos sintácticos y semánticos. Otra de las modificaciones consiste en poder restringir las reglas gramaticales que están activas en cada estado de la comunicación. Es decir, después de que el sistema haya realizado una pregunta al usuario, se pueden restringir las reglas que pueden ser utilizadas para aceptar su respuesta.

El analizador realiza el análisis sintáctico y semántico en paralelo. Cuando las categorías de una regla se han reconocido, se interpretan. Como ya se ha indicado anteriormente, el análisis semántico se basa en el *lambda calculus*. Este formalismo permite una interpretación eficiente y simple. El análisis semántico consiste únicamente en aplicar las funciones lambda sobre los valores lambda en el orden indicado por la regla.

5 El gestor de diálogo

El gestor de diálogo consiste en dos módulos: las reglas de control de diálogo y la representación semántica de la aplicación. La información relativa a la aplicación consiste, como ya se ha descrito en la Sección 2, en una ontología en la que se representan los conceptos de la aplicación. Los atributos de estos conceptos representan la información que se le pedirá al usuario. La información de la aplicación también incluye la descripción de operaciones, que consisten en el identificador y los parámetros.

Las reglas del diálogo no son dependientes de la aplicación, son generales. El diseño de estas reglas sigue el enfoque de actualización del estado de la información (Traum et al. 2000), que supone flexibilizar el modelo de estados finitos. De cada usuario con el que mantiene una conversación, el gestor de diálogo guarda la información de los estados anteriores (la historia). Cuando recibe la interpretación semántica de la última intervención puede así controlar cuál es la información que todavía le falta obtener del usuario concreto.

El gestor de diálogo controla tanto la interacción oral como la escrita. En la comunicación oral los diálogos son totalmente dirigidos por el sistema debido a las limitaciones técnicas. Cuando la entrada es textual se permite al usuario cierta iniciativa, por ejemplo, puede iniciar la conversación aportando cierta información, como en el ejemplo citado *Vivo en la calle Diagonal y quiero tirar una mesa, cuando me la recogerán?*. En este caso, el modelo de actualización del estado de la información debe considerar no sólo el estado anterior sino el resto de estados anteriores (la historia). Se trata de un modelo más adecuado que el sistema de estados finitos, que considera sólo el estado anterior. Siguiendo con el mismo ejemplo, el sistema preguntará al usuario si el objeto es contaminante o no, el nombre y el teléfono del usuario, pero no le preguntará la calle en la que vive, porque esta información ya ha sido previamente introducida por el usuario.

Las reglas del diálogo determinan el siguiente movimiento, es decir, el siguiente atributo el valor del cual debe preguntarse al usuario. El texto que representa la pregunta (o el mensaje) que debe hacerse se ha asociado a cada atributo. También se ha asociado a cada atributo las reglas gramaticales con las que procesar la respuesta del usuario. En el caso de la interacción textual esta información se pasa al servidor de texto. En el caso de la comunicación oral se pasa al intérprete VoiceXML la dirección (el nombre) del fichero que define la interacción para preguntar el valor del atributo. El intérprete de VoiceXML será, de este modo, el encargado de gestionar posibles problemas en la interacción (por ejemplo, el usuario no responde, pide ayuda, el reconocedor no obtiene información aceptable, etc.).

Cuando las reglas del diálogo determinan que se debe realizar un acceso a la aplicación (la información necesaria del usuario ya se ha obtenido) se consulta la descripción de la operación y sus parámetros y se ejecuta. La respuesta obtenida se deberá presentar al usuario. En el caso de la iteración textual se pasará el texto al servidor web y en el caso de la interacción por voz se generará el fichero VoiceXML que determine la respuesta oral del sistema.

6 Conclusiones y trabajo futuro

En este artículo se ha presentado un sistema de diálogos multilingüe basado en la semántica. El sistema permite la interacción oral y textual. El artículo describe con detalle cómo la información de la aplicación puede incorporarse en las gramáticas y léxicos para facilitar el proceso de interpretación semántica. La utilización de una ontología que modele la aplicación permite, además, que la interpretación semántica asociada a las gramáticas y léxicos de diferentes lenguas sea la misma, facilitando así la comunicación en varias lenguas. Por otra parte, el modelado de la aplicación se utiliza también para determinar la lógica que debe controlar la interacción con el usuario.

Actualmente, se esta trabajando en adaptar el sistema a nuevas aplicaciones. También se está trabajando en la incorporación al sistema de las tecnologías de la *Semantic Web*, tanto para representar las ontologías en los estándares más adecuados como para mejorar la interacción con el usuario, especialmente en aplicaciones sobre dominios más complejos, como el servicio de información de las actividades culturales en una ciudad.

Bibliografía

Bateman, J., Magnini, B. and Rinaldi, F. 1994.The Generalized {Italian, German, English} Upper Model. En *proceedings of the ECAI Workshop*.

D'Haro, L.F., Córdoba, R., Ibarz, I., San-Segundo, R., Montero, J.M., Macías-Guarasa, Ferreiros, J., Pardos, J.M. 2004. Plataforma de generación semi-automática de sistemas de diálogo multimodal y multilingüe: Proyecto Gemini. *En las Actas del XX Congreso de la SEPLN.*

Gatius, M.. 2001. "Using an ontology for guiding Natural Language interaction with Knowledge Based Systems". Ph.D. thesis, Softare Department, UPC, 2001.http://www.lsi.upc.es/ ~gatius/tesis.html

Mahesh, K. and Nirenburg, S. 1995. A Situated Ontology for Practical NLP. En *Prcoceedings of IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing,* Montreal, 1995.

Milward, D, Beveridge, M. 2004. Ontologies and the structure od dialogue. En *Prcoceedings of the Eigth Workshop on the Semantics and Pragmatics of Dialogue Catalog*.

Quesada, J. 2002. Modelado de diálogo basado en conocimiento, acciones y expectativas. Procesamiento del Lenguaje Natural, Revista, num.29.

Rodrigo Aguado, L., García Serrano, A., Martínez Fernández, P. 2002. Planeamiento semántico y pragmático para gestión de diálogos en asistentes virtuales. Procesamiento del Lenguaje Natural, Revista, num.28.

Ross, K. 1982.An improved left-corner parsing algorithm. En *Proceedings of the Collin*.

Traum, D., Bos, J., Cooper R., Larson S., Lewin I., Mathesson C., Poesio, M. 2000. A model of Dialogue Moves and Information State Revision. Technical Report D2.1, Trindi Project.

(1) The FADA homepage
http://fada.techideas.info

(2) The HOPS Project.
http://www.hops-fp6.org/

(3) The Loquendo homepage.
http://www.loquendocafe.com/index.asp

(4) W3C Voice Extensible Markup Language. Version 2.0 http://www.w3.org/TR/2004