

Utilización de ontologías en el desarrollo de sistemas de diálogo

Marta Gatus y Meritxell González

Centre de recerca TALP
Universitat Politècnica de Catalunya
gatus@lsi.upc.es

Resumen

En este artículo se propone un diseño para facilitar las tareas de desarrollo de sistemas de diálogo de calidad. Esta propuesta se centra en la representación separada y reutilizable de los diferentes tipos de conocimiento implicados en el proceso comunicativo. Para establecer una clara separación entre las reglas genéricas de diálogo y una aplicación concreta se utiliza una ontología en la que se representan los requerimientos de la aplicación. Las gramáticas y los mensajes del sistema se obtienen utilizando esta ontología y una taxonomía sintáctico-semántica. La metodología propuesta se ha utilizado para desarrollar sistemas de diálogo de voz en VoiceXML y en sistemas de diálogo de entrada textual.

Palabras clave: Sistemas de diálogo, ontologías, reutilización, VoiceXML.

1.- Introducción

El diseño y desarrollo de sistemas de diálogo eficientes, adaptados a la funcionalidad de la aplicación, conlleva un elevado coste. Uno de los principales problemas estriba en los diferentes tipos de conocimiento que deben considerarse: la información que la aplicación requiere del usuario, el modelo de la aplicación que tiene el usuario, la estrategia de comunicación a seguir, el lenguaje que el sistema debe utilizar y reconocer,.... Las diferentes fuentes de conocimiento utilizadas por los sistemas de diálogo son costosas de desarrollar, mantener y reutilizar, requiriendo la intervención de personal experto.

Se han realizado diversos trabajos centrados en un diseño de las diferentes bases de conocimiento que permita reducir el coste del desarrollo de sistemas de diálogo y mejorar su rendimiento. En esta línea, se han realizado diferentes propuestas para representar de forma independiente las reglas genéricas del diálogo y el conocimiento de una aplicación concreta. Estas propuestas incluyen la utilización de una ontología para representar los conceptos de la aplicación ([5]), la definición de un formalismo para especificar diálogos ([6]), el desarrollo de una plataforma que genera diálogos a partir del conocimiento particular de la aplicación ([3]),.... La representación adecuada de las relaciones entre el conocimiento de la aplicación y los recursos lingüísticos que intervienen en su expresión ha sido también abordada en diferentes sistemas de diálogo, como en ([7]).

En este artículo presentamos un diseño que permita mejorar tanto el proceso de comunicación como el desarrollo de diálogos de voz y texto. Esta propuesta se basa en la utilización de una ontología en la que se representan los conceptos de la aplicación que intervienen en el proceso comunicativo. Los recursos lingüísticos para generar las intervenciones del sistema y procesar las del usuario se obtienen utilizando esta ontología y una taxonomía semántico-sintáctica. El diseño propuesto se basa en un trabajo previo centrado en la utilización de ontologías en un tipo concreto de interfaces en lenguaje natural, interfaces que utilizan un sistema de menús mediante el que se muestra al usuario las opciones de lenguaje que el sistema puede aceptar en cada estado ([1]).

En la siguiente sección se describe la utilización de la ontología conceptual en nuestra propuesta, cómo se representa el conocimiento de la aplicación en la ontología y su utilización en las diferentes fases del diseño del diálogo. La aplicación de este diseño para generar diálogos de voz utilizando VoiceXML se presenta en la Sección 3. En la Sección 4 se describe su aplicación para desarrollar un sistema de diálogo de texto. Finalmente, en la última sección se presentan las conclusiones y las líneas de trabajo futuro.

2.- Representación del conocimiento de la aplicación en una ontología

El objetivo primordial de nuestra propuesta consiste en diseñar una organización de los diferentes tipos de conocimiento que permita la fácil adaptación del sistema a nuevas aplicaciones. Para ello, es necesario que el conocimiento específico de una aplicación concreta se represente separadamente del conocimiento genérico necesario para el funcionamiento del diálogo y del conocimiento lingüístico que interviene en su expresión. Nuestra propuesta aborda este problema mediante la utilización de una ontología en la que se representará el conocimiento específico de cada aplicación. Esta ontología se utilizará en las diferentes fases del desarrollo del sistema de diálogo: diseño de la estructura lógica de la interacción entre el sistema y el usuario, generación de las intervenciones del sistema y generación de las gramáticas necesarias para procesar las intervenciones del usuario. A continuación se describe la utilización de la ontología en estas distintas fases.

2.1. Diseño de la estructura lógica de la interacción entre el sistema y el usuario

En la ontología conceptual se representarán los conceptos de la aplicación que intervienen en la comunicación. Cada uno de estos conceptos se describe mediante un conjunto de atributos. El proceso comunicativo girará entorno al valor de estos atributos. El sistema de diálogo deberá obtener el valor de algunos de los atributos de los conceptos a partir de las intervenciones del usuario. Los valores de estos atributos se utilizarán para realizar las operaciones necesarias sobre la aplicación con los parámetros adecuados. Las respuestas del sistema a las preguntas del usuario consistirán en la expresión del valor de algunos de los atributos. El orden en que la información relativa a los diferentes conceptos se pregunta al usuario puede representarse en la ontología en forma de precondiciones.

Consideremos, por ejemplo, el caso de una aplicación en la que deba realizarse una transacción. En general, este tipo de aplicaciones ofrecen tres servicios diferentes: dan información, realizan la transacción y permiten cancelar una transacción previa. Estas aplicaciones pueden representarse en la ontología mediante una clase general que representa la aplicación y tres subclases que representan los correspondientes servicios de información, transacción y cancelación. La clase general estará descrita, como mínimo, por un atributo que indica el tipo de servicio que desea el usuario. La Figura 1 muestra esta representación genérica para las aplicaciones que realizan transacciones. Las tres subclases que representan los tres posibles servicios incluyen la siguiente precondición:

atributo_valor (transacción, tiposervicio, Servicio)

donde la variable *Servicio* tendrá como valor el identificador (nombre) que corresponde al servicio.

Esta precondición indica que solamente se realizará el servicio en cuestión si así lo indica el valor del atributo *tiposervicio* del concepto **Transacción**, es decir, si el usuario lo ha expresado en una intervención previa. Por ejemplo, sólo se le pedirá información relativa a una cancelación si previamente ha indicado que desea hacer una cancelación.

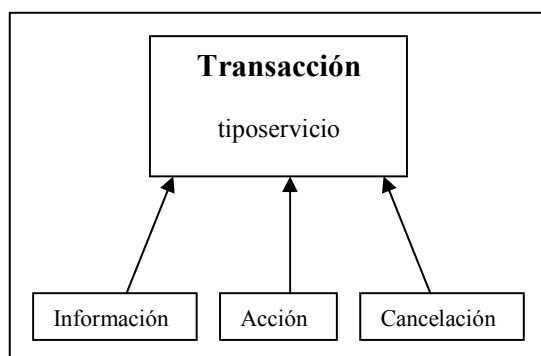


Figura 1: Representación de aplicaciones de transacción

Además de esta información genérica, los conceptos transacción, información, acción y cancelación incorporan información específica de cada aplicación. El concepto que representa la acción a realizar se describe mediante los atributos que representan los parámetros necesarios para realizar la correspondiente operación sobre la aplicación (i.e., nombre del usuario, teléfono, dirección,...).

La estructura lógica de las interacciones vendrá determinada por la representación el sistema de diálogo, no obstante, diferentes tipos de diálogo podrán obtenerse a partir de una misma representación: dirigidos por el sistema, de iniciativa mixta o de iniciativa de usuario (en el caso de entrada textual).

La representación de la información relativa a la aplicación en la ontología se obtiene estudiando el modelo de las tareas de la aplicación y el conocimiento que de esas tareas tiene el usuario. La información que sobre la aplicación tiene el usuario puede inferirse de muestras de diálogos. Estos ejemplos permiten conocer la dificultad que conlleva para el usuario comprender la información que una aplicación necesita, si el usuario dispone o no de esta información, etc. A modo de ejemplo, analicemos uno de los problemas que aparece en el sistema de diálogo que hemos desarrollado para una aplicación de recogida de muebles y electrodomésticos de casas particulares. La aplicación necesita información sobre si el objeto que se debe recoger es contaminante o no. El estudio de muestras de diálogo nos permitió comprobar que en algunos casos el usuario desconoce si un objeto es contaminante o no. Por ello, es mejor incorporar este tipo de conocimiento en la ontología como una taxonomía en la que se describen los objetos que deben ser recogidos.

La ontología no se utiliza únicamente para determinar la información que la aplicación necesita del usuario y viceversa, el sistema de diálogo la puede utilizar también para realizar preguntas de clarificación. En el caso en el que una respuesta del usuario sea un hiperónimo o hipónimo del valor esperado, el sistema de diálogo puede, accediendo a la ontología, procesar la respuesta y realizar la pregunta que le permita obtener el dato que necesita. En el caso de la aplicación de la recogida de objetos mencionada anteriormente, podríamos encontrar el siguiente ejemplo:

S: ¿Que tipo de objeto quiere tirar?

U: Un electrodoméstico

S: ¿Qué tipo de electrodoméstico?

U: Una nevera

En este ejemplo, el sistema espera un tipo de mueble específico (una nevera, una lavadora,...) y, sin embargo, recibe como respuesta un término más genérico. La representación de la taxonomía de objetos en la ontología, permite al sistema deducir que la respuesta del usuario no corresponde a ninguno de los objetos esperados pero si a uno de sus hipónimos y, por tanto, podrá utilizar la pregunta adecuada para obtener el valor que necesita.

2.2. Generación del conocimiento lingüístico

La mayoría de los sistemas de diálogo utilizan texto “enlatado” (previamente introducido) en las diferentes intervenciones del sistema: mensajes que guían al usuario a

introducir la información necesaria, preguntas de clarificación o confirmación, respuestas,... El problema principal que presenta la utilización de texto predefinido es que en la fase de desarrollo se deben de obtener y guardar todas las posibles intervenciones que pueden ser necesarias en cada uno de los diferentes estados del diálogo.

En la propuesta que presentamos los textos utilizados en las intervenciones del sistema pueden ser obtenidos de forma automática a partir de la representación de la aplicación en la ontología. Para ello se utiliza una taxonomía semántico-sintáctica, en la que se clasifican todos los atributos que describen los conceptos. Las diferentes clases de esta taxonomía corresponden a los diferentes roles gramaticales que aparecen en el sublenguaje utilizado en los diálogos estudiados.

La idea de una base de conocimiento semántico-sintáctica, que actúe como interfaz entre el conocimiento conceptual representado en una ontología y los recursos léxico-gramaticales, ha sido desarrollada en otros trabajos, siendo uno de los más reconocidos el General Upper Model, descrito en [2]. En nuestra propuesta, la base semántico-sintáctica se limita a los atributos de conceptos cuyo valor aparece en las conversaciones hombre-máquina.

Cada una de las clases de la taxonomía semántico-sintáctica está asociada con un conjunto de plantillas que representan diferentes formas que el sistema puede utilizar para preguntar o responder sobre el valor de los atributos en esas clases. En estas plantillas deberán incorporarse adecuadamente las entradas léxicas correspondientes al concepto, al atributo y a sus valores. Para que este proceso pueda realizarse de forma automática es necesario que en la ontología se incorporen estas entradas léxicas.

Consideremos, por ejemplo, la aplicación de recogida de objetos mencionada anteriormente. Se trata de una aplicación en la que debe realizarse una transacción y, por tanto, se representará en la ontología el concepto *Recogida* como un elemento de la clase *Transacción*, mostrada en la Figura 1. Siguiendo esta representación el concepto *Recogida* consta de tres servicios: información, petición (la acción en este caso consiste en concertar un día de recogida) y cancelación. Además el concepto *Recogida* está descrito por el atributo *tiposervicio*. El atributo *tiposervicio* se ha representado también como una clase en la taxonomía sintáctico-semántica, que pertenece a la clase más general *tipo*. La plantilla por defecto asociada a esta clase *tiposervicio* es la siguiente:

“Este servicio permite obtener información sobre su funcionamiento, <verbo_acción> y cancelar una acción anterior. Qué es lo que desea?”,

en donde *<verbo_acción>* debe sustituirse por la entrada léxica correspondiente al grupo verbal que expresa la acción a realizar, en este caso sería *“concertar un día de recogida”*.

La clase *tiposervicio* tiene también asociada una segunda plantilla que permite preguntar de forma más directa la misma información. Esta segunda plantilla puede utilizarse, por ejemplo, cuando la respuesta del usuario a la pregunta anterior no haya podido ser procesada y deba repetirse. La plantilla es: *“Qué desea? Información,<nombre_acción> o cancelación.”*

en donde *<nombre_acción>* representa el nombre asociado a la acción, en el ejemplo descrito sería *“recogida de objetos”*.

El mecanismo para la generación de mensajes es sencillo, se instanciarán las plantillas generales asociadas a cada uno de los atributos cuyo valor pueda ser preguntado o explicado al usuario durante la comunicación.

El desarrollo de gramáticas y léxicos que permitan procesar las diferentes intervenciones del usuario suele ser un proceso costoso, que requiere la participación de expertos. La mayoría de estas gramáticas incorporan información semántica que facilita la obtención de los datos que la aplicación requiere, por lo que son totalmente dependientes de la aplicación.

En nuestra propuesta, las gramáticas y léxicos pueden obtenerse también a partir del conocimiento de la aplicación representado en la ontología. El procedimiento es similar al que se ha descrito para generar los mensajes del sistema. Dado que las intervenciones del usuario se limitan a describir (o preguntar) el valor de un atributo, las gramáticas necesarias para una aplicación concreta son aquellas que representan las diferentes frases con las que se pueden expresar los valores de los atributos conceptuales que aparecen en la ontología.

Cada una de las clases de atributos en la taxonomía semántico-sintáctica se asocia con las reglas gramaticales que reconocen las diferentes frases con las que se expresan los valores de los atributos. A partir de estas reglas generales se obtienen las gramáticas particulares para cada aplicación. Los lexicones incorporarán un conjunto de entradas léxicas comunes a todas las aplicaciones y otro formado por los grupos nominales y verbales asociados a conceptos, atributos y valores de la aplicación.

La taxonomía de atributos es fácilmente reutilizable en diferentes lenguas. Aunque esta clasificación se ha desarrollado considerando las distinciones en castellano, la mayoría de distinciones aparecen en otras lenguas. La misma clasificación se ha utilizado, por ejemplo, para obtener recursos lingüísticos en catalán.

Una de las principales ventajas de la metodología propuesta es que permite generar las gramáticas y los mensajes de sistema en diferentes lenguas a partir de la misma representación conceptual. Otra de las ventajas de obtener los recursos lingüísticos de forma semi-automática a partir de la representación de la aplicación es que la interrelación y consistencia entre las diferentes fuentes de conocimiento es mayor. No obstante, los recursos generados deben ser revisados y adaptados a las necesidades particulares de cada aplicación.

3.- Utilización del método en el desarrollo de sistemas de diálogo en VoiceXML

El desarrollo del lenguaje estándar VoiceXML ([11]) para crear sistemas de diálogos ha revolucionado las aplicaciones comerciales basadas en la voz. VoiceXML, basado en el lenguaje XML, permite un desarrollo rápido de los sistemas orales para información en la Web. No obstante, la funcionalidad de la mayoría de sistemas comerciales en VoiceXML es a menudo reducida. En muchos casos, este pobre funcionamiento no es sólo originado por las

limitaciones técnicas, sino también por un diseño de diálogo mejorable. Por esta razón, la mejora del diseño de los sistemas desarrollados en VoiceXML ha sido objeto de estudio tanto desde el ámbito comercial ([4]), como académico ([10]). En esta sección se describe cómo el diseño general para desarrollar sistemas de diálogos descrito en la sección anterior puede utilizarse para facilitar la creación de diálogos en VoiceXML.

Un sistema VoiceXML para una aplicación concreta consiste en un conjunto de documentos, en los que se describen las interacciones con el usuario. El resultado de una interacción lleva a la siguiente, por lo que un sistema VoiceXML puede definirse como una máquina de estados finitos. Los documentos VoiceXML incorporan, por cada interacción, los mensajes del sistema, la gramática que debe utilizarse para reconocer la respuesta del usuario y la siguiente acción a realizar por cada posible respuesta del usuario. Las acciones que el sistema puede realizar finalizada una interacción incluyen la repetición de mensajes, el control de una nueva interacción, el acceso a la aplicación, etc.

```
<?xml version="1.0"?>
<vxml version="2.0" xml:lang="es-es">
  <form id="formATTRNAME">
    <field name="attrATTRNAME">
      <grammar src="file://gramATTRNAME.sjv"
        type="application/x-jsgf-flx"/>
      <!-- PromptAskATTRNAME-->
      <prompt count = 1>
        PROMPTATTRNAME1
      </prompt>
      <prompt count = 2>
        PROMPTATTRNAME2
      </prompt>
      <filled>
      <if
        cond="attrATTRNAME=='VAL1ATTRNAME'" >
        <goto next="VAL1ATTRNAME.vxml"/>
      <elseif
        cond="attrATTRNAME=='VAL2ATTRNAME'"/>
        <goto next="VAL2ATTRNAME.vxml"/>
      <elseif
        cond="attrATTRNAME=='VAL3ATTRNAME'"/>
        <goto next="VAL3ATTRNAME.vxml"/>
      </if>
      </filled>
    </field>
  </form>
</vxml>
```

Figura 2: Un documento de VoiceXML genérico

Nuestra propuesta consiste en utilizar la estrategia descrita en la Sección 2 en el diseño y desarrollo de sistemas de diálogo en VoiceXML. La utilización de una ontología en la que se representa el conocimiento de la aplicación que aparece en la comunicación nos ha permitido desarrollar un procedimiento que genera automáticamente los documentos VoiceXML necesarios para una aplicación determinada. Para cada atributo conceptual cuyo valor debe obtenerse durante la comunicación

(representado en la ontología), el procedimiento genera un documento que define la interacción en la que se le pide al usuario su valor. Los documentos VoiceXML se generan a partir de un documento genérico, que se va copiando, sustituyendo adecuadamente la información relativa a un atributo concreto: el nombre del atributo, las preguntas del sistema para obtener su valor y las gramáticas. La Figura 2 muestra el documento genérico que define la interacción para preguntar el valor de un atributo. El documento generado a partir de este documento genérico para el atributo *tiposervicio* del concepto *Recogida* (descrito en la sección anterior) se representa en la Figura 3.

```
<?xml version="1.0"?>
<vxml version="2.0" xml:lang="es-es">
  <form id="formtiposervicio">
    <field name="attrtiposervicio">
      <grammar src="file://gramtiposervicio.sjv"
        type="application/x-jsgf-flx"/>
      <!-- PromptAsktiposervicio-->
      <prompt count = 1>

      Este servicio permite obtener información sobre su
      funcionamiento, concertar un día de recogida y
      cancelar una acción anterior. Qué es lo que desea?

      </prompt>
      <prompt count = 2>

      Qué desea? Información, recogida de objetos o
      cancelación

      </prompt>
      <filled>
      <if
        cond="attrtiposervicio=='informacion'" >
        <goto next="informacion.vxml"/>
      <elseif
        cond="attrtiposervicio=='recogida'"/>
        <goto next="recogida.vxml"/>
      <elseif
        cond="attrtiposervicio=='cancelacion'"/>
        <goto next="cancelacion.vxml"/>
      </if>
      </filled>
    </field>
  </form>
</vxml>
```

Figura 3: El documento para preguntar el tipo de servicio

La ventaja de utilizar un documento genérico para este proceso es que el documento puede ser fácilmente reemplazable. Los diálogos generados a partir del documento genérico de la Figura 2 son simples y completamente dirigidos por el sistema. Actualmente, se está trabajando en la mejora del procedimiento para que pueda también generar diálogos de una cierta complejidad, en los que el usuario tenga cierta iniciativa (por ejemplo, el orden en que expresa el valor de algunos atributos).

Tanto el mensaje(s) del sistema como la gramática que permite obtener la respuesta del usuario se genera utilizando la estrategia descrita en la sección anterior. Siguiendo esta estrategia, los atributos que describen los conceptos se deben clasificar según la taxonomía semántico-sintáctica. Las entradas léxicas correspondientes al concepto, al atributo y a sus valores se han de incorporar en la ontología. Los mensajes del sistema se obtienen a partir de las plantillas asociadas a las clases de los atributos, incorporando adecuadamente las entradas léxicas correspondientes al atributo y al concepto.

Las gramáticas utilizadas por nuestro sistema son muy simples. En la Figura 4 se muestra la gramática utilizada para procesar la respuesta del usuario cuando, utilizando el documento mostrado en la Figura 3, se le pide el tipo de servicio que desea. Como puede verse en la figura, la gramática consta de tres reglas, cada una de las cuales reconoce uno de los servicios que ofrece la aplicación. La información semántica asociada a cada una de las reglas corresponde a un posible valor del atributo *tiposervicio* del concepto *Recogida*.

```
public <gramtiposervicio> = ( <GramInf1>{:ret} |
<GramC1>{:ret} | <GramT1>{:ret} )
    {<@gramtiposervicio $ret>};

<GramInf1> = ( información )
    {return("INFORMACION")};

<GramC1> = ( cancelar )
    {return("CANCELACION")};

<GramT1> = ( [concertar un día] recogida [de
objetos] ) {return("RECOGIDA")};
```

Figura 4: La gramática que reconoce el tipo de servicio

4. Utilización del método en el desarrollo de diálogos de texto

El rápido crecimiento de la Web y la consiguiente cantidad de información y servicios accesibles, abre nuevos horizontes para las diferentes tecnologías de información. Entre otros retos que ha planteado esta nueva situación, ha surgido el de guiar al usuario en la búsqueda de información. Se plantea la necesidad de asistentes virtuales que guíen al usuario a acceder a la información que una determinada empresa o institución dispone en la Web. La mayoría de asistentes virtuales existentes no incorporan procesamiento de lenguaje natural, en muchos casos utilizan solamente patrones de pregunta/respuesta. No obstante, actualmente se están desarrollando asistentes que incorporan técnicas desarrolladas para los sistemas de diálogo de entrada textual ([8]), mejorando de esta forma el proceso interactivo.

Actualmente, estamos trabajando en el desarrollo de un sistema de diálogo capaz de guiar al usuario en el acceso a los servicios e información disponibles en un portal concreto de la

Web. Con este fin, estamos desarrollando un sistema que sea fácilmente adaptable a diferentes dominios y servicios, siguiendo el diseño descrito en la primera sección.

Las reglas del diálogo que controlan las diferentes interacciones son genéricas y, por tanto, independientes del dominio. El diseño de estas reglas sigue el enfoque de actualización del estado de la información ([9]). Este enfoque se basa en la actualización dinámica del estado de información, representado en una estructura semántica. El modelo de actualización del estado de la información supone flexibilizar el modelo de estados finitos. En el modelo de estados finitos se definen todas las posibles transiciones del diálogo, a partir del estado actual y de las diferentes posibles intervenciones del usuario. El modelo de los estados finitos se ha utilizado en muchos diálogos de voz porque permite utilizar un modelo de lenguaje (gramática) adaptado a cada estado. En estos sistemas la iniciativa del diálogo corresponde al sistema. El modelo de actualización del estado de la información considera no sólo el estado anterior sino el resto de estados anteriores (la historia) y permite el diseño de diálogos de iniciativa mixta.

En los sistemas de entrada textual, dado que las limitaciones técnicas para reconocer la entrada del usuario son mucho menores, la iniciativa suele ser mixta o, incluso, puede ser del usuario. Por ello, en nuestro sistema se han incorporado también elementos de la teoría de acción conjunta, en la que se asume que el sistema y el usuario colaboran para obtener la información necesaria. Siguiendo este modelo, en nuestro sistema se incorporan subdiálogos de ayuda, de clarificación, confirmaciones, etc.

En el sistema de diálogo que estamos desarrollando, la ontología en la que se representan los requerimientos de la aplicación se utiliza tanto para obtener la lista de estados posibles (en los que el sistema deberá pedir determinada información), como para generar diálogos de clarificación o ayuda cuando la respuesta del usuario no es la esperada. En esta primera etapa del desarrollo hemos implementado un prototipo muy simple, en el que tanto las gramáticas utilizadas para procesar las intervenciones del usuario como los mensajes del sistema son generados de forma semi-automática a partir de la representación de los requerimientos del sistema en la ontología.

Se utilizan gramáticas libres de contexto aumentadas con rasgos sintácticos y semánticos que facilitan su interpretación. Las reglas gramaticales incorporan información semántica basada en *lambda calculus*. Esta información consiste en una lista de números que indica el orden en que deben interpretarse los constituyentes una vez reconocidos. Las reglas gramaticales pueden incluir precondiciones que controlan su activación, de manera que en un determinado estado del diálogo se pueden activar o desactivar. Se utiliza un analizador de *charts* basado en unificación. Este analizador realiza el procesamiento sintáctico y semántico de forma paralela. Tanto el controlador de diálogo como el analizador se han implementado en lenguaje Prolog.

5. Conclusiones y trabajo futuro

En este artículo se ha presentado una propuesta para facilitar las tareas del diseño y desarrollo de sistemas de diálogo de calidad. Esta propuesta se basa en representar de forma separada los diferentes tipos de conocimiento que intervienen en la comunicación: las reglas genéricas que determinan la interacción, los conceptos de la aplicación que intervienen en el diálogo y los recursos lingüísticos relacionados con estos conceptos. Esta organización facilita el correcto diseño de sistemas de diálogo, así como la reutilización de sus componentes, ya que sólo el conocimiento estrictamente relacionado con la aplicación debe representarse en la ontología. La utilización de una ontología en la que se representan los conceptos utilizados en la aplicación puede favorecer también la calidad de la comunicación. Se ha comprobado su utilidad para detectar hiperónimos e hipónimos del valor que se espera obtener de la intervención del usuario. Por otra parte, la organización del conocimiento propuesta facilita el desarrollo de sistemas que soportan varias lenguas, porque el conocimiento lingüístico y el conocimiento conceptual están representados de forma independiente.

La estrategia propuesta se ha seguido en el desarrollo de sistemas de diálogo en VoiceXML. Concretamente, se ha utilizado para desarrollar un sistema de diálogos para una aplicación que permite concertar fechas de recogida de objetos. La utilización de esta metodología en aplicaciones prácticas nos ha permitido comprobar que su utilización favorece el desarrollo de herramientas que permiten automatizar la creación de los componentes del sistema que dependen de la aplicación: documentos VoiceXML, mensajes del sistema, gramáticas,... El trabajo futuro en ésta línea incluye la construcción de herramientas que nos permitan incorporar una mayor complejidad en los componentes generados automáticamente: documentos VoiceXML que contengan interacciones de iniciativa mixta, gramáticas que puedan ser modificadas dinámicamente, gramáticas para reconocer diversas formas de expresión,...

El diseño propuesto se ha utilizado también en un asistente virtual para la Web que actualmente se está desarrollando. El sistema se está desarrollando para guiar al usuario en la utilización de un servicio que permite concertar una recogida de muebles desde la Web. En un futuro el sistema se utilizará también para guiar al usuario en la búsqueda de información relativa a una determinada entidad disponible en la Web.

Uno de los problemas que presenta la utilización de ontologías es que deben incorporarse las especificaciones de cada aplicación, además de conocimiento del dominio. Ello requiere de una persona que conozca cómo se debe representar el conocimiento relacionado con la aplicación en la ontología. Nuestro trabajo futuro incluye el estudio de mejoras que permitan facilitar esta tarea, como la incorporación de metaconceptos que recojan el funcionamiento de diferentes tipos de aplicaciones, como el concepto *Transacción*, descrito en la Sección 2. El trabajo futuro también incluirá el estudio de nuevas mejoras en el proceso comunicativo que pueden derivarse de la utilización de la ontología, además de las ya descritas.

6. Referencias

- [1] Gatiús, M. and Rodríguez, H. Natural Language Guided Dialogues for Accessing the Web. En las Actas del 5th International Conference, Text, Speech and Dialogue 2002. Brno, Czech Republic, September 2002.
- [2] Bateman, J., Magnini, B. and Rinaldi, F. The Generalized {Italian, German, English} Upper Model. En las Actas de ECAI Workshop, 1994.
- [3] D'Haro, L.F., Córdoba, R., Ibarz, I., San-Segundo, R., Montero, J.M., Macías-Guarasa, Ferreiros, J., Pardos, J.M. Plataforma de generación semi-automática de sistemas de diálogo multimodal y multilingüe: Proyecto Gemini. En las Actas del XX Congreso de la SEPLN, Barcelona 2004.
- [4] Loquendo. <http://www.loquendocafe.com/index.asp>
- [5] Milward, D, Beveridge, M. Ontologies and the structure od dialogue. En las Actas del Eighth Workshop on the Semantics and Pragmatics of Dialogue Catalog. Barcelona, 2004
- [6] Quesada, J. F. Modelado de diálogo basado en conocimiento, acciones y expectativas. Procesamiento del Lenguaje Natural, Revista, num.29. Septiembre de 2002.
- [7] Pérez-Piñar López, García Mateo. Integración automática de fuentes de conocimiento lingüístico en el desarrollo de sistemas de diálogo. Procesamiento del Lenguaje Natural, Revista, num.29. Septiembre de 2002.
- [8] Rodrigo Aguado, L., García Serrano, A., Martínez Fernández, P. Planeamiento semántico y pragmático para gestión de diálogos en asistentes virtuales. Procesamiento del Lenguaje Natural, Revista, num.28. Mayo de 2002.
- [9] Traum, D., Bos, J., Cooper R., Larson S., Lewin I., Mathesson C., Poesio, M. A model of Dialogue Moves and Information State Revision. Informe Técnico D2.1, Trindi Project.
- [10] Villarejo, L., Castell, N. and Hernando, J. Dialogue Management in an Automatic Meteorological Information System. International Conference IEA/AIE 2003 (Industrial and Engineering Applications on Artificial Intelligence and Expert Systems). June, 23-26, 2003, LNAI vol. 2718, pp. 495-504. Loughborough, England.
- [11].W3C Voice Extensible Markup Language. Version 2.0 <http://www.w3.org/TR/2004>