Talking Agents: Arquitectura para Sistemas de Agentes Conversacionales

José María Fernández de Alba López de Pablo
<jmfernandezdalba@gmail.com>

Director:
Dr. Juan Pavón Mestras
<jpavon@fdi.ucm.es>

Proyecto Fin de Máster en Sistemas Inteligentes
Máster en Investigación en Informática, Facultad de Informática,
Universidad Complutense de Madrid
Curso 2008 - 2009

El/la abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: "Talking Agents: Arquitectura para Sistemas de Agentes Conversacionales", realizado durante el curso académico 2008-2009 bajo la dirección de Dr. Juan Pavón Mestras en el Departamento de Ingeniería del Software e Inteligencia Artificial (ISIA), y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

José María Fernández de Alba López de Pablo

Resumen

Los Talking Agents son entidades software con la capacidad de reconocer el habla humana y sintetizar una respuesta hablada. Se han desarrollado este tipo de agentes para construir instalaciones artísticas, con el propósito de trabajar en nuevos tipos de experiencias en la interacción del espectador con la obra de arte. El diseño e implementación de los talking agents afronta varios problemas por sus requisitos de rendimiento y alto grado de configurabilidad para múltiples escenarios con distintos tipos de recursos en juego. Una forma de abordar estos requisitos ha sido la distribución de los componentes de cada agente y una clara separación entre agentes y recursos. El trabajo describe la arquitectura de los talking agents a distintos niveles, y algunos escenarios de experimentación de los mismos.

Palabras clave. Talking agent, arquitectura sistema multiagente, reconocimiento y síntesis de habla, instalación artística.

Abstract

Talking agents are software entities with the ability to recognize human speech and synthesize a spoken response. We have developed this kind of agents for building installation-art works, with the purpose of work in new kinds of experiences in the interaction of the spectator with the art work. The design and implementation of talking agents confronts several issues because of their requirements on performance and high degree of configuration for multiple scenarios with different kinds of resources in play. One way to cope with these requirements has been the distribution of the components of each agent and a clear separation between agents and resources. This work describes the architecture of talking agents at different levels, and several experimentation scenarios with them.

Keywords. Talking agent, multi-agent system architecture, speech recognition and synthesis, installation-art.

Índice general

Capítulo 1

Introducción

En el terreno del arte, la mayoría de las obras basadas o inspiradas en el mundo tecnológico digital consisten apenas en la producción de efectos visuales o en el movimiento de dispositivos mecánicos como resultado de la interacción del espectador, siguiendo algoritmos o patrones simples. Aún no existen demasiadas aplicaciones en este campo de los avances en las técnicas de inteligencia artificial, las cuales podrían soportar interacciones a un nivel cognitivo superior, para concebir nuevas formas de confrontación del espectador frente al trabajo artístico. En particular, este proyecto considera el uso de técnicas de procesamiento de voz y del lenguaje como base para la interacción entre el espectador y la obra, la cual sucede en un contexto que puede reforzarse con las habilidades sociales de los sistemas multiagente.

El bloque de construcción básico para la instalación artística es el Talking Agent, una entidad autónoma y reutilizable con la habilidad de interactuar con humanos mediante la voz, que existe en un entorno computacional distribuído. Una instalación consistirá en varios Talking Agent, organizados como una sociedad de agentes que pueden cooperar y afectarse mutuamente. Los Talking Agent permiten explorar problemas conceptuales desde la perspectiva del arte, tales como el test de Turing o el comportamiento social emergente en la sociedad digital, donde los humanos y las entidades artificiales viven juntos.

1.1. Motivación

1.1.1. Artística

Desde un punto de vista artístico, el motivo del trabajo es la realización de una instalación artística compuesta por una serie entidades con las que un espectador pueda interactuar de diversas maneras, y principalmente mediante el lenguaje natural hablado. Esto pretende explorar distintas formas de acceder a la sensibilidad artística del espectador, introduciéndolo en un entorno social con el que interactúa a un nivel cognitivo mayor que con otro tipo de trabajos de arte y computación.

1.1.2. Técnica

Desde un punto de vista técnico, el motivo del trabajo es la puesta a prueba de manera conjunta en un entorno de agentes software, de diversas técnicas de inteligencia artificial, sobre todo los sistemas de razonamiento basados en reglas y basados en casos; y de ingeniería lingüística, incluyendo técnicas de análisis y síntesis de lenguaje natural y tecnologías de reconocimiento y síntesis de voz.

Sin embargo, otro motivo quizá más interesante es el de la búsqueda de formas alternativas de análisis del lenguaje natural, aparte de las bien conocidas, como el parsing estadístico, para abordar situaciones de posible gran imprecisión sintáctica. Para esta tarea se ha hecho uso de inspiración proveniente del campo de la recuperación de información (information retrieval).

1.2. Objetivos del Trabajo

El objetivo global del trabajo consiste en conseguir un sistema basado en agentes en el que cada uno de ellos, además de comunicarse con el resto, sea capaz de mantener una conversación hablada con un espectador sobre un tema arbitrario, pudiendo definir el "guión" a seguir por cada agente. Además, se pretende enriquecer la interacción agregando un determinado "carácter" a cada uno, que modifique su manera de expresarse.

Aparte de lo anterior, el sistema ha de ser modificable en cada punto de su funcionamiento, pudiendo tanto agregar nuevas interacciones de diversas naturalezas, como modificar la manera en que se interpretan los eventos y la manera en que se generan las respuestas.

Para ello se definen los siguientes objetivos concretos:

- definición y el desarrollo de una arquitectura de agentes conversacionales, que sea lo suficientemente flexible como para poder admitir la agregación de nuevos medios de interacción de distinta naturaleza y de distinto nivel de abstracción, como sensores del entorno, comunicación hablada, interacción física, etc

- definición y desarrollo de un sistema de Natural Language Understanding (NLU) que sea capaz de entender discursos de cualquier naturaleza, haciendo uso de sistemas de razonamiento basados en casos

- definición y desarrollo de un sistema de control y decisión para controlar las acciones llevadas a cabo por el agente según sus creencias sobre el entorno, haciendo uso de máquinas de estados y sistemas de razonamiento basados en reglas

- definición y desarrollo de un sistema de Natural Language Generation (NLG) simple que permita generar discursos diferentes teniendo en cuenta un carácter atribuído a cada agente

1.3. Estructura de la Memoria

La estructura de la Memoria es como sigue:

Capítulo 1. Esta introducción.

Capítulo 2. Se describen otras arquitecturas de agentes conversacionales y sistemas de gestión de diálogo y las tecnologías involucradas en ellas.

Capítulo 3. Se detalla cómo se organiza y funciona un agente como sistema de diálogo y cada una de sus partes.

Capítulo 4. Se detalla cuál es la arquitectura multiagente propuesta y cómo se relacionan agentes y recursos.

Capítulo 5. Se muestran una serie de escenarios de uso de los Talking Agents y la discusión de los resultados.

Capítulo 6. Se dan una serie de conclusiones y observaciones sobre el trabajo y las líneas alternativas y futuras de desarrollo.

Capítulo 2

Estado del Arte

A continuación se va a hacer un breve repaso de los distintos campos relacionados con este trabajo en el ámbito científico-técnico. Hay que tener en cuenta que, dado que estos campos están relacionados con la lingüística, gran parte de su aplicación está supeditada al grado de independencia del lenguaje que se haya considerado, o al grado de desarrollo en distintos lenguajes que exista. Muchas de las elecciones de tecnologías para utilizar en el sistema han estado influídas por este motivo.

Los principales campos considerados en el trabajo con los siguientes: Arquitecturas de Sistemas de Diálogo, Análisis de Lenguaje Natural, Síntesis de Lenguaje Natural, Herramientas de Reconocimiento del Habla y Herramientas de Síntesis de Voz.

2.1. Arquitecturas de Sistemas de Diálogo

Para alcanzar el objetivo de la interacción con las máquinas a través del lenguaje natural, tal y como lo hacemos con las personas, se ha realizado mucha investigación para desarrollar arquitecturas que puedan combinar las distintas tecnologías necesarias en este contexto, tales como el reconocimiento de voz, análisis de lenguaje natural, generación de lenguaje natural, síntesis de voz, etc. de una forma genérica, extensible y mantenible. Muchas de estas arquitecturas además pretenden ser *multimodales*, i.e. ser capaces de representar tipos adicionales de interacción bajo un esquema integrado.

Los primeros intentos de arquitecturas de este tipo consistían en un simple ”pipelining” de la información a través de distintos niveles de abstracción. Las fases principales de este ”pipeline” eran: percepción de la entrada, análisis a distintos niveles de la entrada para determinar la acción, gestión y decisión sobre la interacción, generación de la respuesta y ejecución de la respuesta. Este es aproximadamente el enfoque en [7] (figura 2.1), usado para un sistema de consultas sobre horarios de trenes, y aún se pone como ejemplo para este tipo de sistemas por su simplicidad.

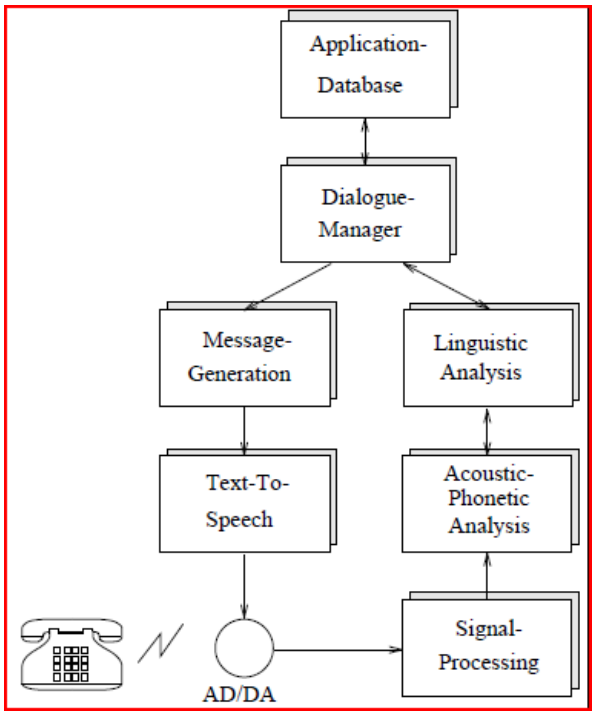

Figura 2.1: Arquitectura en [7]

Una arquitectura más elaborada para este propósito es la descrita en [2], para distintos tipos de sistemas, la cual identifica explícitamente algunos elementos como el gestor de contexto del discurso, gestor de referencias y el planificador de contenido (figura 2.2). Esta arquitectura es mejorada posteriormente en [3], describiendo una organización más clara en tres secciones: interpretación, generación y comportamiento. (figura 2.3).

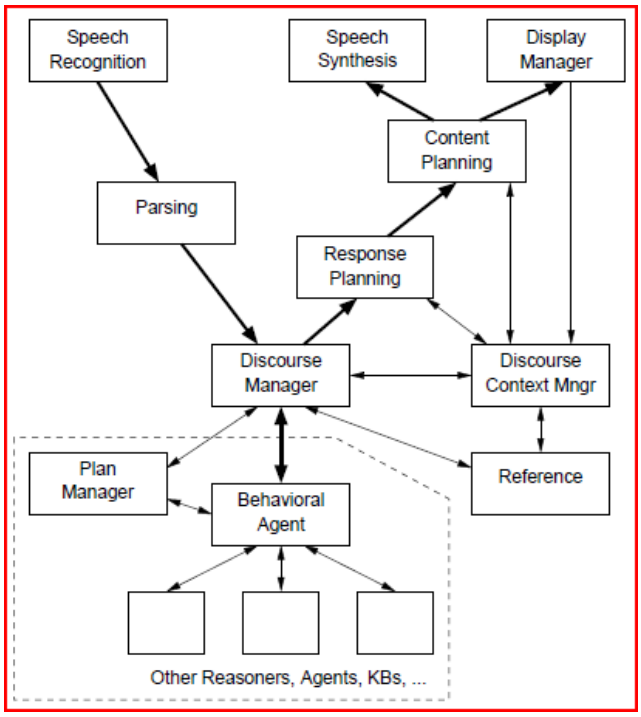

Figura 2.2: Arquitectura en [2]

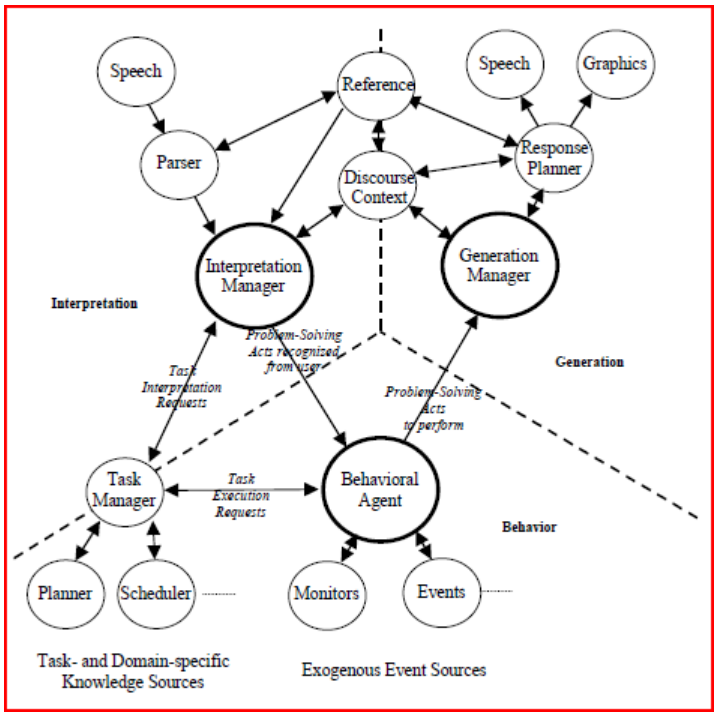

Figura 2.3: Arquitectura en [3]

Un paso más allá en la definición de arquitecturas considera la distinción entre comunicación *a nivel de interacción* y comunicación *a nivel de contenido*. La comunicación a nivel de interacción se refiere a la comunicación en sí misma y se usa para gestionar ciertos parámetros como la velocidad, el turno, la forma, etc de la comunicación. La comunicación a nivel de contenido se refiere al propio propósito de la interacción. Un ejemplo de una arquitectura que sigue esta idea se describe en [24].

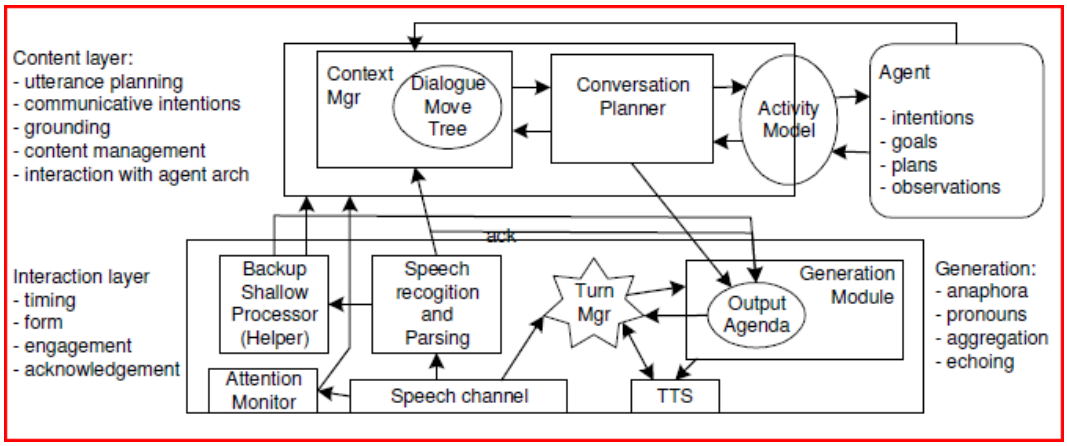

Figura 2.4: Arquitectura en [24]

Otros enfoques lo que hacen es mejorar la arquitectura básica descrita al principio con algunos elementos para rellenar ciertos huecos o debilidades en lugar de crear una nueva arquitectura de la nada. Esto es lo que se hace en [5], donde la arquitectura trata con el tema de la comunicación a nivel de interacción usando un módulo de interpretación mejorado que es capaz de discriminar entre eventos de interacción y de contenido (figura 2.5).

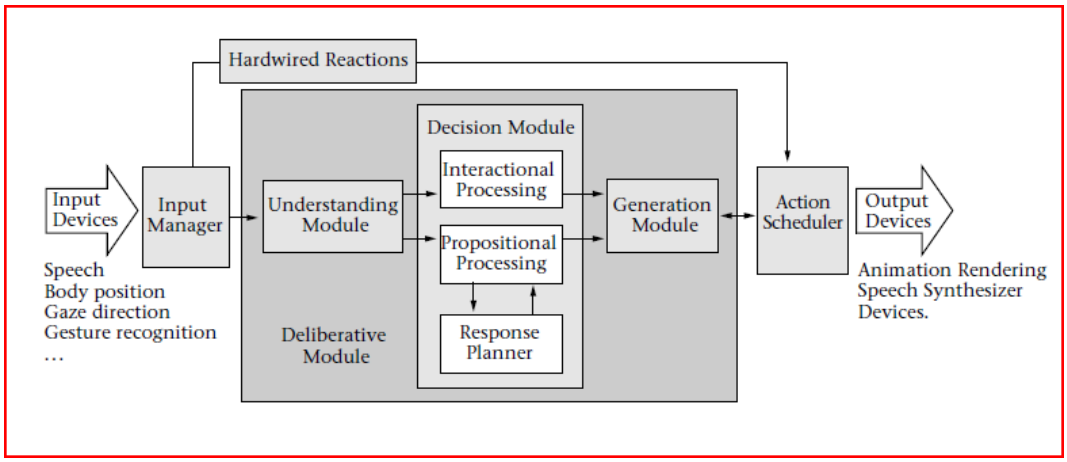

Figura 2.5: Arquitectura en [5]

Además, también hay otras arquitecturas que tienen en cuenta otros aspectos aparte de la gestión del diálogo, por ejemplo la capacidad de distribución de los componentes, separando los componentes de tal manera que puedan ser usados como un servicio. Esta idea es comentada en [9] como ejemplo, y se utiliza como punto de partida en este trabajo (figura 2.6).

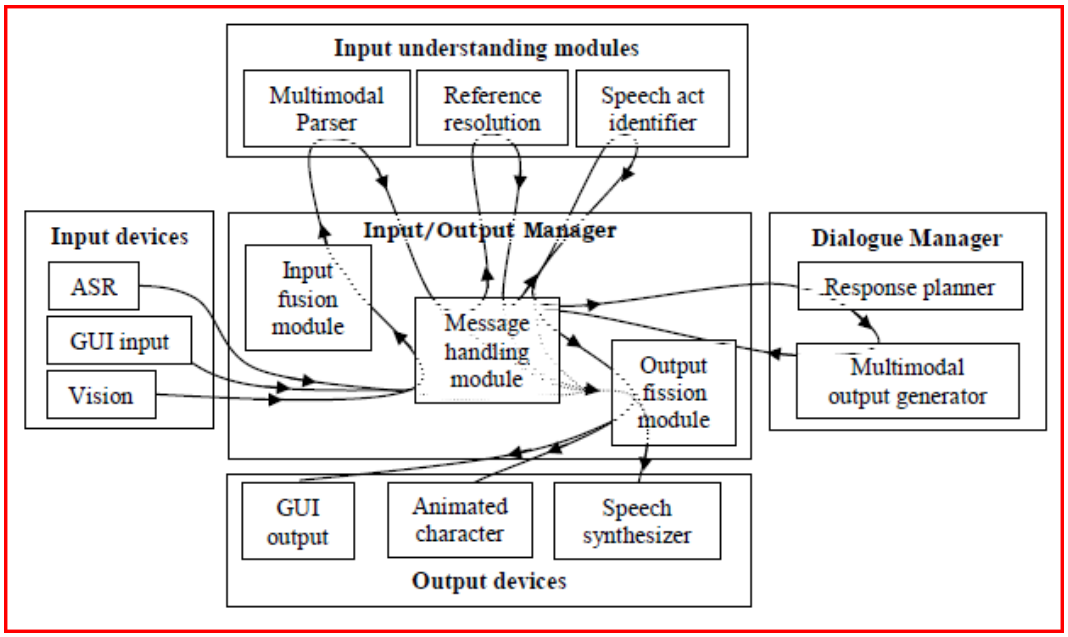

Figura 2.6: Arquitectura en [9]

2.2. Análisis de Lenguaje Natural

El análisis del lenguaje natural, también conocido como "natural language parsing" o "natural language understanding" [1] es un proceso equivalente a la compilación de un lenguaje de programación. Se trata de, dada una cadena de caracteres, extraer una estructura sintáctica y semántica que la represente computacionalmente. En el caso del lenguaje natural, el objetivo es conseguir realizar el análisis sintáctico de una frase y posteriormente relacionar sus elementos con el contexto para conseguir un determinado conocimiento.

Como el lenguaje natural tiende a ser impreciso tanto sintáctica como semánticamente, no suele hacerse uso de los tradicionales analizadores basados en gramáticas que usan los compiladores. En su lugar se utilizan analizadores estadísticos, los cuales tienen en cuenta las probabilidades de los distintos tipos de construcciones posibles, y escogen aquellos más probables. Algunos ejemplos de herramientas de análisis son el Stanford Parser [20] en inglés, y FreeLing [11] en español, las cuales son capaces de generar árboles sintácticos a partir de frases en lenguaje natural.

Relacionado con este tema, existen además otros tipos de analizadores, como EmoTag [8] que, dado un texto en lenguaje natural, es capaz de asignar etiquetas

emocionales que representan las emociones que se evocan en el texto. En el contexto de este trabajo, éste tipo de análisis puede ser útil para tener en cuenta el estado de ánimo del espectador a la hora de razonar.

En este trabajo se ha optado por un análisis basado en sistemas de razonamiento basados en casos, que trata de extraer la información de un discurso en lenguaje natural comparándolo con otros casos almacenados anteriormente. El motivo de esta decisión es que la irregularidad de los discursos previstos como entrada, principalmente consultas sobre el futuro, dificulta que el conocimiento se extraiga de una manera más ordenada.

2.3. Síntesis de Lenguaje Natural

La síntesis del lenguaje natural, o "natural language generation" consiste en generar una salida en lenguaje natural para comunicar cierta información. Este proceso puede ser muy sencillo o muy complejo dependiendo de la complejidad de la información que se quiere comunicar. En general, para generar una salida de cualquier complejidad, son necesarias ciertas fases de planificación de contenido, que incluyen agregación de oraciones [21], resolución de referencias [22] y la elección de las palabras concretas a utilizar [4]. Un ejemplo de esto se encuentra en [23].

Siguiendo estas fases, se consigue un proceso totalmente genérico e independiente del idioma objetivo concreto, salvo en las últimas fases, en las que se deben elegir las palabras específicas a usar. Además, permite llevar a cabo construcciones sintácticas complejas, como por ejemplo oraciones subordinadas, para lograr representar las posibles relaciones entre los elementos de información que desea comunicar.

En el caso de este trabajo, se ha optado por una opción basada en sistemas de razonamiento basados en casos, que principalmente tan sólo lleva a cabo un proceso de resolución de referencias después de elegido el contenido. Este método se ha elegido para poder introducir más fácilmente una mayor variedad en los textos generados al no estar restringidos a un proceso de generación más definido.

2.4. Herramientas de Reconocimiento del Habla

Actualmente, las tecnologías de reconocimiento del habla están en un punto de desarrollo en el que se consiguen unas tasas de reconocimiento lo suficientemente altas como para llevar a cabo sin demasiados problemas, aplicaciones basadas en la interacción mediante habla. Sin embargo debido a la naturaleza de este tipo de tecnologías, dependientes del idioma, este desarrollo no se encuentra en el mismo punto en todos los casos, siendo en este caso el inglés el idioma que marca la cabeza del desarrollo.

La tasa de reconocimiento de un determinado sistema está directamente relacionada con el tamaño del dominio de palabras o construcciones sintácticas en el que trabaje. Así, pueden distinguir varios tipos de sistemas de reconocimiento según su generalidad o especificidad en determinados dominios sintácticos. Normalmen-

te, estos sistemas utilizan una representacion de dicho dominio, llamada *modelo de lenguaje*, que contiene la información necesaria para reconocer palabras dentro de ese dominio. Estos modelos de lenguaje pueden obtenerse de manera automática mediante herramientas como el Statistical Language Modeling Toolkit [17], a partir de un corpus de texto lo bastante amplio.

También afecta a la tasa de reconocimiento el grado de ”adaptación” de la herramienta a la voz de una persona en particular. Hay herramientas que aprenden a reconocer con mayor precisión la voz de una determinada persona, sin embargo, esta cualidad no interesa especialemente en este trabajo, ya que con en sistema va a interaccionar una gran variedad de usuarios.

Hay que aclarar, que el desarrollo de este tipo de herramientas, no entra dentro del alcance de este trabajo, y por tanto no se pretende llevar a cabo investigación en este campo, sino tan sólo conocerlo lo suficiente como para poder elegir una herramienta existente que se adecúe a los requisitos.

De entre todos los sistemas de reconocimiento aplicables a este trabajo, se ha restringido la búsqueda a aquellos que implementan la interfaz definida por Java Speech API [13], por motivos de implementación, al ser una interfaz de Java y ser de uso bastante extendido. Implementaciones de esta interfaz, en la parte de reconocimiento, son por ejemplo Sphinx-4 [16] y TalkingJava [18]. La primera, basada en el uso de modelos de lenguaje creados con el SLMT (mencionado anteriormente), permite una mayor personalización del dominio, sin embargo, existe la traba de la falta de desarrollo libre en corpus y lexicones del español, necesarios como entrada para el SLMT. Por otra parte, la segunda, basada en Microsoft Speech API [15], permite tan sólo una personalización muy limitada, y está restringida a su uso en un entorno Windows. Sin embargo, la segunda opción ofrece un reconocimiento del español de dominio general muy aceptable para los objetivos de este trabajo y por ese motivo ha sido la elegida.

2.5. Herramientas de Síntesis de Voz

Al igual que con los sistemas de reconocimiento de voz, el desarrollo de los sistemas de síntesis de voz está marcado por su avance en el inglés. Este tipo de herramientas hacen uso de bases de datos de dialófonos, i.e. sonidos que pueden articularse para formar palabras, y de otro tipo de reglas lingüísticas de descomposición en sílabas, correspondencia entre sílabas y sonidos, reglas de acentuación, etc, que son completamente dependientes del idioma, y por tanto limitan la utilización de las herramientas. Además, hay que tener en cuenta el factor estético y de inteligibilidad de los sonidos, lo que también limita el hacer uso indiscriminado de sonidos sintéticos para la creación de estas bases de datos.

Por el mismo motivo que la tecnología anterior, el desarrollo de este tipo de herramientas no entra dentro del alcance de este trabajo, y por tanto tan sólo se pretende conocer el campo lo suficiente como para poder elegir una herramienta existente que se adecúe a los requisitos.

La falta de definiciones libres de modelos fonéticos en español, dificulta el uso de herramientas de síntesis genéricas (i.e. que permiten personalizar su función a

un idioma concreto, especificando los modelos fonéticos a utilizar), como FreeTTS [12], la cual tiene la ventaja añadida de implementar la Java Speech API [13], que es una interfaz en Java de uso bastante extendido.

Por estos motivos, se han explorado otros tipos de sistemas, incluídos de software propietario, con un buen soporte para el español, como TextSound [19] o Loquendo [14], decidiéndose finalmente el uso del primero, únicamente por motivos "estéticos".

2.6. Conclusiones

Tras los análisis previos, se ha decidido que los campos donde merece más la pena investigar en este trabajo son los tres primeros, dado que es ahí donde se puede admitir una mayor variabilidad con la que obtener distintos resultados interesantes. Los dos últimos, en cambio son campos donde el desarrollo ha llegado a niveles difícilmente mejorables y por este motivo, es preferible hacer uso de los desarrollos existentes e integrarlos en el sistema. Las elecciones concretas en cada campo se han especificado en su sección correspondiente.

Capítulo 3

Arquitectura del Talking Agent

A continuación se explica cuál es la estructura y funcionamiento de los Talking Agents como agente conversacional y el de cada una de sus partes. Para ello, en principio se detalla la visión estática del sistema, es decir, las relaciones entre sus partes, y posteriormente se describe la visión dinámica, donde se explica cómo interaccionan esas partes para llevar a cabo su cometido.

3.1. Visión Estática

3.1.1. Visión General del Sistema

En la figura 3.1 se muestran las partes que conforman al Talking Agent como agente conversacional. Cada una de éstas está desacoplada de las demás y puede estar distribuída en un nodo diferente. De este modo el Talking Agent queda como un mero gestor que coordina cada uno de los recursos a su disposición, usando el modelo de control reactivo (ver Framework ICARO 4.1), y el módulo de decisión, para determinar qué iniciativas emprender en cada momento.

Los distintos tipos de recursos: Percepción, Interpretación, Planificación y Ejecución; le permiten gestionar una interacción de cualquier naturaleza (multimodal) de una manera completamente genérica, como se explicará más adelante.

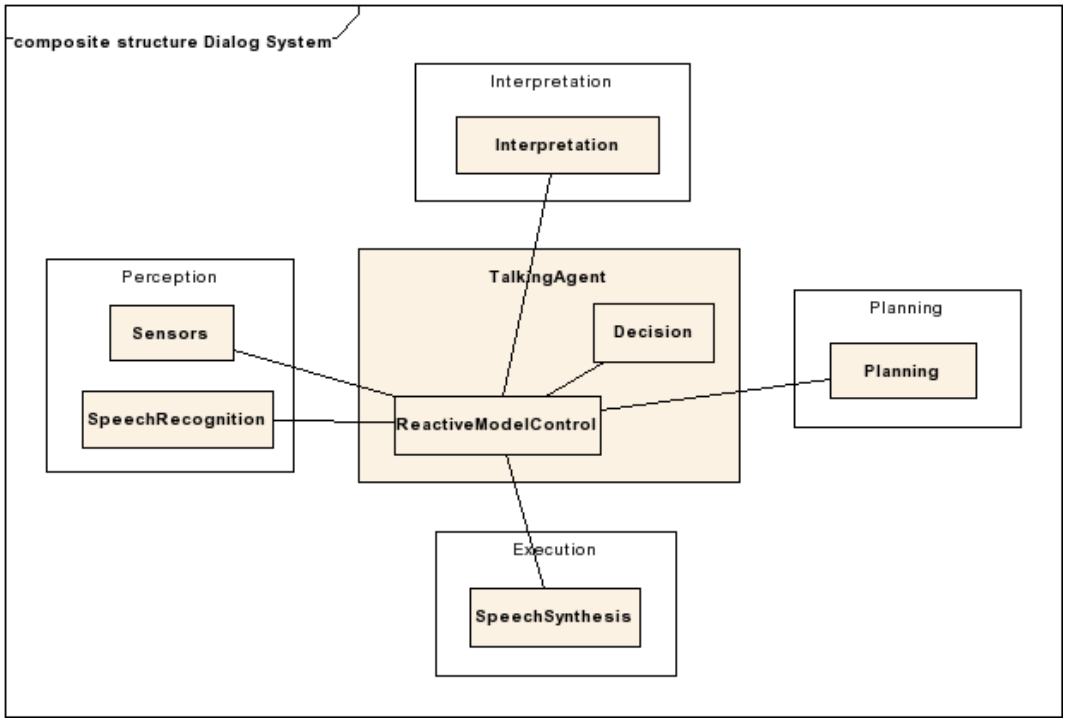

Figura 3.1: Visión general del sistema

3.1.2. Elementos del Sistema

Estos son los distintos módulos involucrados en el sistema, descritos separadamente.

Percepción

La percepción se define como un thread que genera eventos cuando detecta aquellas situaciones para las que está preparado. En el caso de la percepción de sensores cuando un sensor es activado y en el caso de la percepción de discurso, cuando se escucha un discurso. Su estructura puede observarse en la figura 3.2.

La clase Perception y sus subclases heredan de la clase Thread. Esto es así porque la percepción siempre ha de ejecutarse de manera asíncrona al control que la gestiona, de tal manera que pueda percibir el exterior y generar los eventos apropiados a sus Oyentes.

La percepción de Texto es un caso particular de percepción, y la percepción de Discurso es un caso particular de esta última. Cada tipo de percepción genera un tipo de evento, que deberá ser manejado por los Oyentes suscritos a cada uno de ellos.

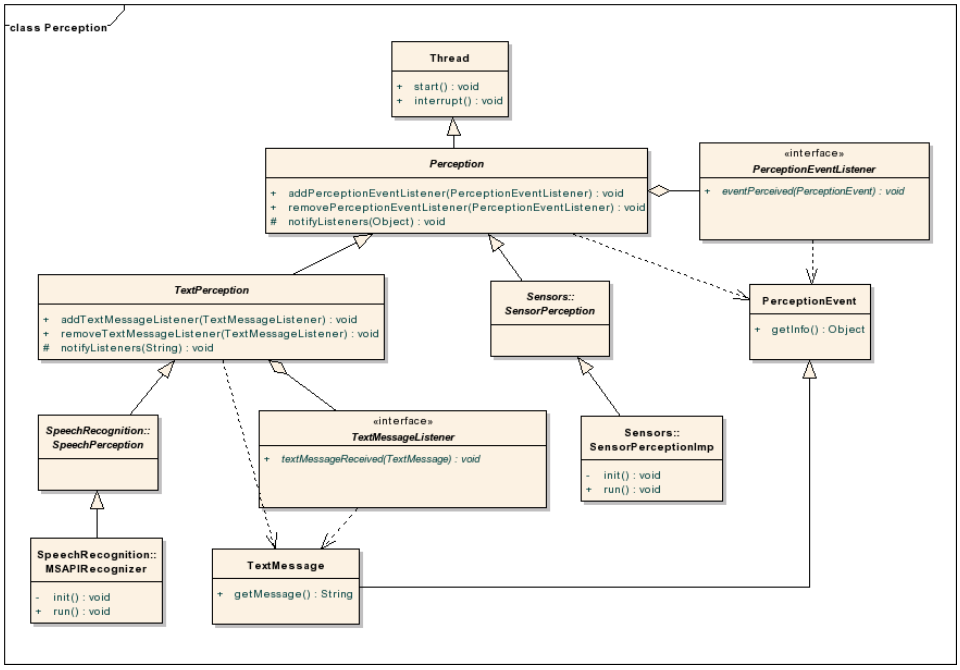

Figura 3.2: Estructura de la Percepción

Interpretación

La interpretación se lleva a cabo en dos niveles: uno específico y otro global. En el específico se interpretan los eventos de una naturaleza concreta, por ejemplo, los eventos de sensores, los eventos de voz, los eventos de interfaz, etc. En el global, se tienen en cuenta las interpretaciones de cada uno de los eventos específicos, para poder obtener una interpretación más completa, si es necesario.

En este caso, hay dos tipos de intérpretes específicos: el de los eventos de los sensores y el de los eventos de voz. El intérprete global es capaz de determinar a qué tipo de intérprete específico encomendar cada tipo de evento.

Los eventos de los sensores son muy simples ya que sólo hay un tipo de sensor y su activación siempre tiene el mismo significado.

Por otra parte, los eventos de voz se interpretan haciendo uso de un sistema de razonamiento basado en casos, en concreto JColibri2 [6]. Para ello se utiliza una descripción de la situación actual con la que se hace una búsqueda de casos que se asemejen a dicha situación, los cuales ofrecerán una solución adecuada. La estructura de los casos se define más adelante.

Este tipo de implementación para los eventos de voz se ha elegido por su flexibilidad y por la naturaleza de las interpretaciones que han de hacerse, sin embargo, se podría haber hecho uso de una interpretación en la que interviniesen análisis sintácticos y semánticos más profundos de las frases recibidas, haciendo uso de las técnicas descritas en 2.2.

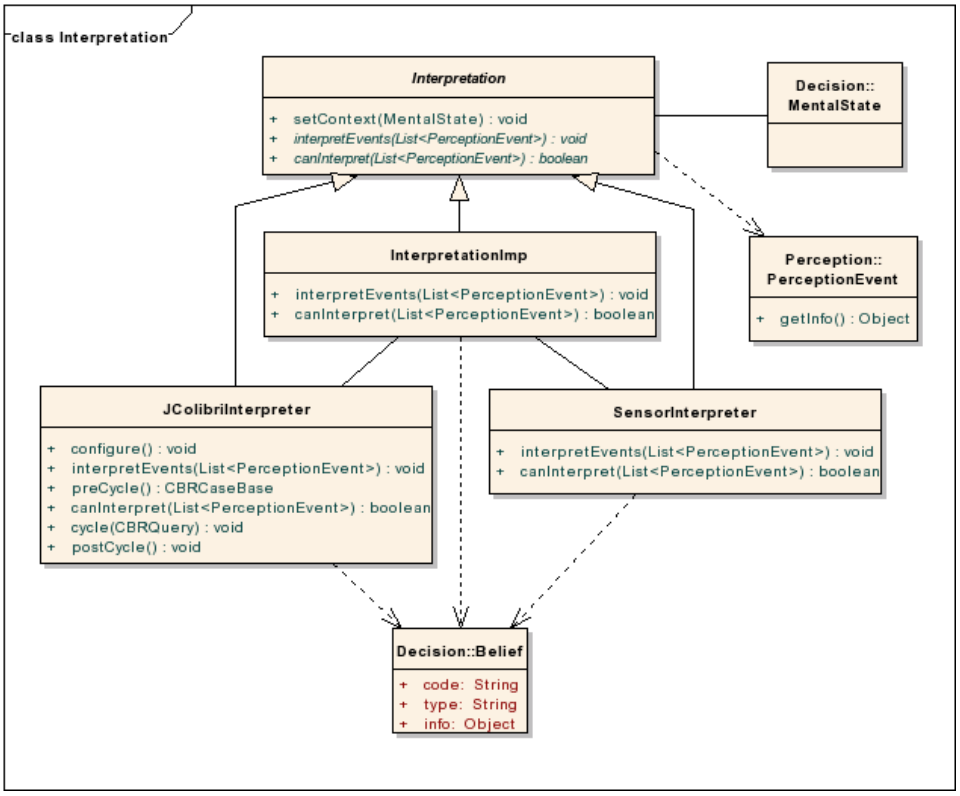

Figura 3.3: Estructura de la Interpretación

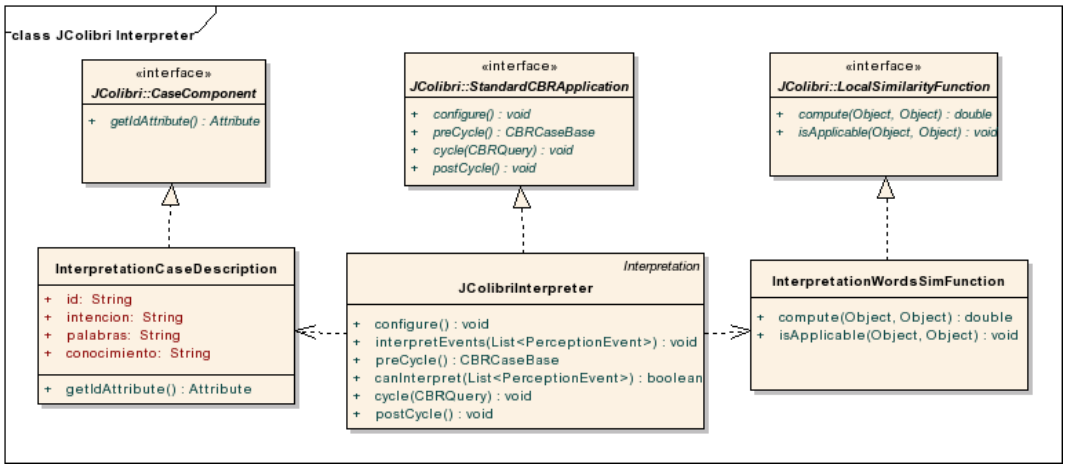

Figura 3.4: Detalle JColibriInterpreter

Esta es la estructura de los casos:

Descripción del Caso

- **Intencionalidad Esperada.** es un identificador que representa lo que se espera que diga el visitante. Lo que se espera que diga depende de qué le ha dicho el agente previamente. Así, si el agente le pregunta cuál es su nombre, inmediatamente se espera que el visitante responda eso mismo. Esta intencionalidad esperada se determina en la fase de Decisión, en el momento en el que el agente decide una acción, inmediatamente prevee sus "consecuencias". Estos son unos ejemplos de intencionalidades:

  - responderTema → la intencionalidad esperada cuando se le pregunta sobre el tema de su consulta
  - responderProblema → la intencionalidad esperada cuando se le pregunta sobre el problema concreto que tiene
  - responderConfirmacion → la intencionalidad esperada cuando se le pide confirmar algo
  - ...

- **Palabras.** es una lista de palabras separadas por espacios que están contenidas en lo que de hecho ha dicho el visitante. Todas las variantes de género, número, etc de las palabras se consideran una misma palabra. Por ejemplo:

  ```
  padre hospital medico muerte
  ```

  Esta descripción coincide con un discurso en el que se hayan mencionado todas esas palabras. Sin embargo, para poder condensar casos, además se pueden poner listas adicionales en las líneas subsiguientes, de tal manera que la descripción coincide si el discurso contiene todas las palabras de al menos una de las listas. Por ejemplo:

  ```
  padre hospital medico muerte
  cancer terminal
  ```

  Esta descripción coincide con un discurso que contenga todas las palabras de la primera línea o que contenga todas las palabras de la linea siguiente.

Solución del Caso

- **Conocimiento Adquirido.** es una lista de identificadores separados por comas y precedidos por una palabra clave, que representa el tipo de conocimiento adquirido. Este conocimiento se almacena en la memoria del agente en forma de "creencias", que representan principalmente el contexto de la conversación. Éstos son algunos tipos:

- **tema:** palabra clave que indica que los próximos identificadores son el tema o temas del discurso. Los identificadores són como los siguientes: amor, salud, trabajo, amistad, etc.
- **problema:** palabra clave que indica que los próximos identificadores son el problema o problemas concretos que se citan en el discurso. Los identificadores serán como los siguientes: amorCompartido, muerteFamiliar, despido, etc. Debe existir una relación entre cada uno de estos identificadores y los de tema, de tal manera que cada identificador de problema pertenezca a un tema, por ejemplo:
  - amor
    - amorCompartido, celos, ruptura, ...
  - salud
    - cancer, enfermedadCronica, ...
  - trabajo
    - despido, nuevoTrabajo, ...
- **confirmacion:** palabra clave que indica que el próximo identificador es un sí o un no. Se utiliza para reconocer respuestas afirmativas o negativas.

Estos son algunos casos de ejemplo:

Descripción	
Intencionalidad Esperada	responderTema
Palabras	salud enfermo hospital fiebre seria
Solución	
Conocimiento Adquirido	tema salud

Descripción	
Intencionalidad Esperada	responderProblema
Palabras	coche estrellar moto estrellar accidente trafico choque
Solución	
Conocimiento Adquirido	problema coche

Descripción	
Intencionalidad Esperada	responderConfirmacion
Palabras	si por supuesto
Solución	
Conocimiento Adquirido	confirmacion si

Descripción	
Intencionalidad Esperada	responderConfirmacion
Palabras	no ni hablar
Solución	
Conocimiento Adquirido	confirmacion no

Decisión

Esta es la parte donde el agente coordina cada uno de los módulos a su disposición y su ciclo de funcionamiento y decide qué camino tomar dependiento del conocimiento que tenga sobre su situación y la del entorno. La decisión se lleva a cabo a dos niveles: a nivel de control y a nivel de intención.

En la figura 3.5 se muestran las clases que componen el módulo de decisión, lo que conforma el núcleo del Talking Agent.

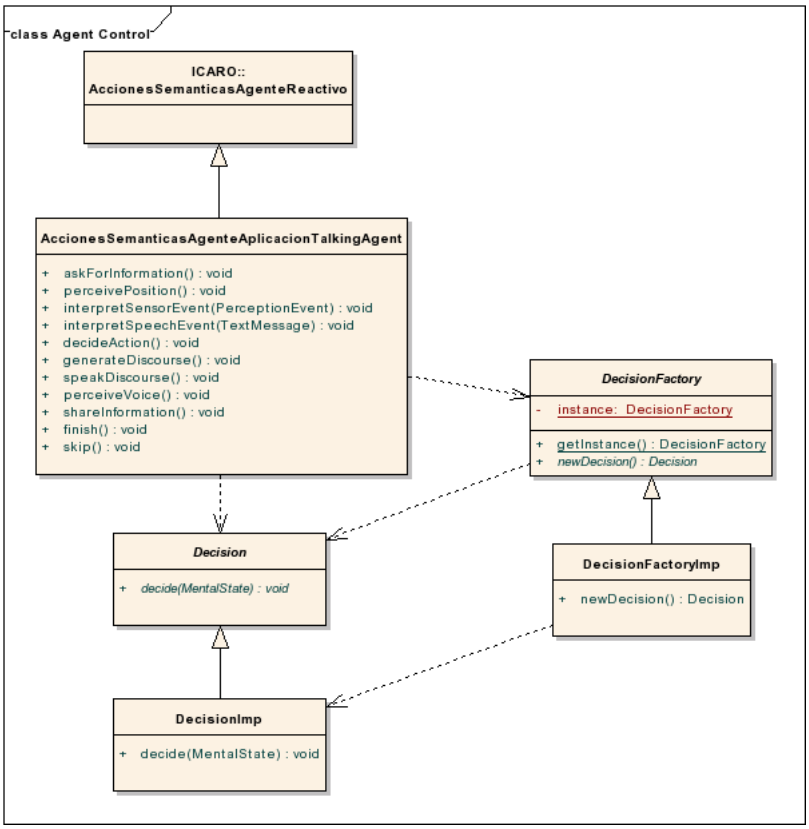

Figura 3.5: Estructura del Control del Agente

Nivel de Control A este nivel de decisión, el agente tan sólo decide qué fase del flujo de información de la interacción va a procesar a continuación. Éstas decisiones se modelan en forma de máquina de estados, que controla cada una de las funciones llevadas a cabo por el agente en cada momento, tal y como muestra la figura 3.6. En ICARO, esta máquina de estados se representa textualmente mediante un fichero XML como el mostrado en el cuadro 3.1

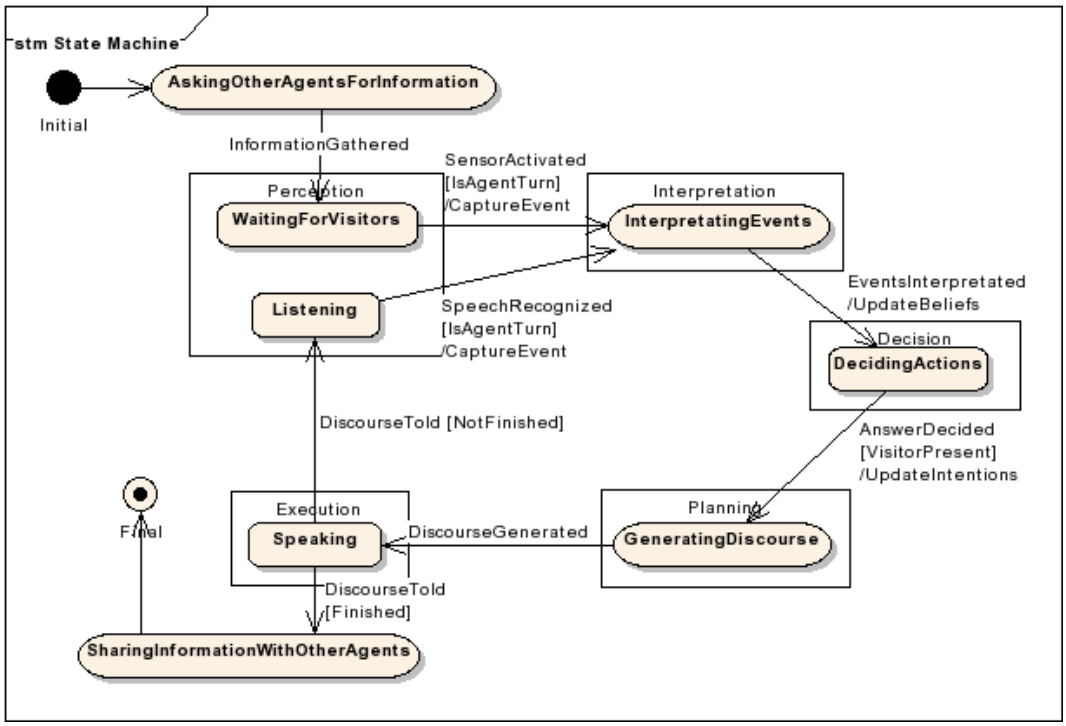

Figura 3.6: Máquina de estado de Control

```
<?xml version="1.0"?>
<!DOCTYPE tablaEstados SYSTEM "schemas/TablaEstados.dtd">
<tablaEstados descripcionTabla="Statechart for the Talking Agent">
    <estadoInicial idInicial="initialState">
        <transicion input="comenzar"
                accion="askForInformation"
                estadoSiguiente="askingOtherAgentsForInformation"
                modoDeTransicion="bloqueante"/>
    </estadoInicial>
    <estado idIntermedio="askingOtherAgentsForInformation">
        <transicion input="informationGathered"
                accion="perceivePosition"
                estadoSiguiente="waitingForVisitors"
                modoDeTransicion="bloqueante"/>
    </estado>
    <estado idIntermedio="waitingForVisitors">
        <transicion input="sensorActivated"
                accion="interpretSensorEvent"
                estadoSiguiente="interpretatingEvents"
                modoDeTransicion="bloqueante"/>
    </estado>
    <estado idIntermedio="interpretatingEvents">
        <transicion input="eventInterpretated"
                accion="decideAction"
                estadoSiguiente="decidingActions"
                modoDeTransicion="bloqueante"/>
    </estado>
    <estado idIntermedio="decidingActions">
        <transicion input="answerDecided"
                accion="generateDiscourse"
                estadoSiguiente="generatingDiscourse"
                modoDeTransicion="bloqueante"/>
    </estado>
    <estado idIntermedio="generatingDiscourse">
        <transicion input="discourseGenerated"
                accion="speakDiscourse"
                estadoSiguiente="speaking"
                modoDeTransicion="bloqueante"/>
    </estado>
    <estado idIntermedio="speaking">
        <transicion input="discourseTold"
                accion="perceiveVoice"
                estadoSiguiente="listening"
                modoDeTransicion="bloqueante"/>
        <transicion input="finished"
                accion="shareInformation"
                estadoSiguiente="sharingInformationWithOtherAgents"
                modoDeTransicion="bloqueante"/>
    </estado>
    <estado idIntermedio="listening">
        <transicion input="speechRecognized"
                accion="interpretEvent"
                estadoSiguiente="interpretatingEvents"
                modoDeTransicion="bloqueante"/>
    </estado>
    <estado idIntermedio="sharingInformationWithOtherAgents">
        <transicion input="informationShared"
                accion="skip"
                estadoSiguiente="initialState"
                modoDeTransicion="bloqueante"/>
    </estado>
    <estadoFinal idFinal="finalState"/>
    <transicionUniversal input="termina" accion="finish" estadoSiguiente="finalState"
            modoDeTransicion="bloqueante"/>
    <transicionUniversal input="error" accion="clasificaError" estadoSiguiente="finalState"
            modoDeTransicion="bloqueante"/>
</tablaEstados>
```

Cuadro 3.1: Fichero XML con la máquina de estados

Nivel de Intención A este nivel, el agente decide cuáles son sus próximas intenciones a llevar a cabo. Aquí, el conocimiento usado obvia detalles irrelevantes tales como las palabras concretas usadas por el interlocutor, la frecuencia de su voz, etc. Con la misma, las decisiones generadas también obvian esos detalles, expresando sólo las intenciones del agente desde un punto de vista abstracto. Los detalles concretos dependen de los niveles inferiores.

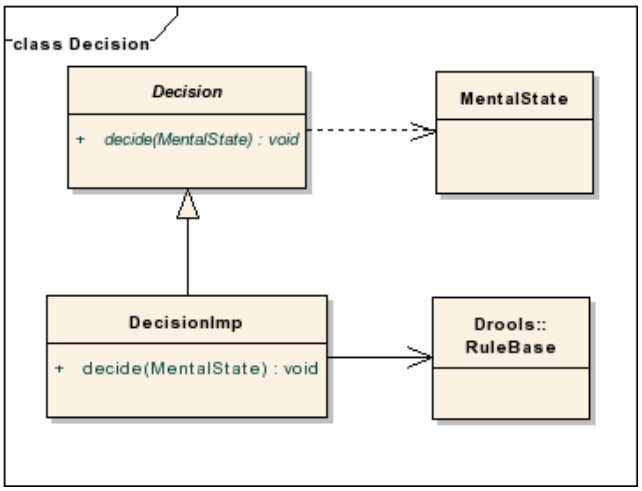

Figura 3.7: Estructura de la Decisión

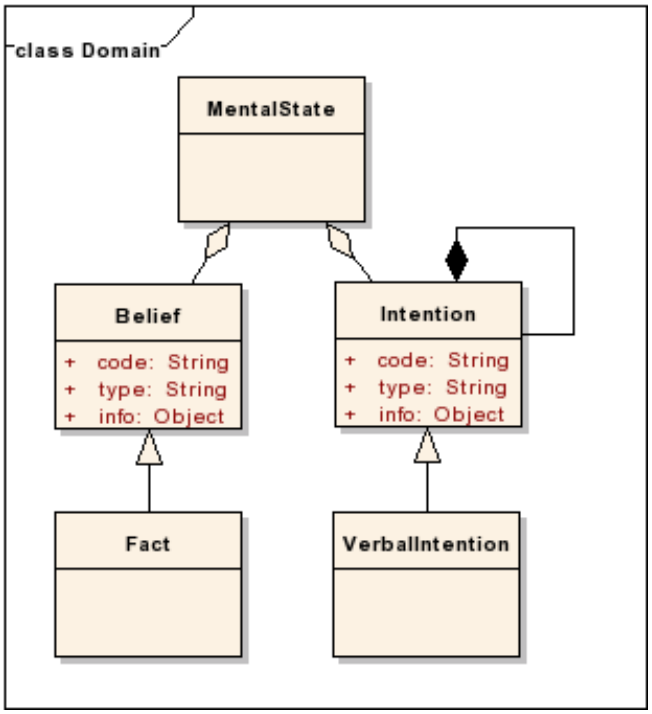

Figura 3.8: Estructura del Estado Mental

La elección de la decisión depende de los hechos y creencias que se encuentren en el estado mental del agente: cada agente intenta hacer cosas basándose en lo que sabe y lo que cree. A partir de ahí, se utilizan una serie de reglas en el formato Drools [10] que determinan de qué manera cada situación concluye en una intención determinada. Las reglas tienen la forma:

```
package icaro.aplicaciones.agentes.agenteAplicacionTalkingAgentReactivo.comportamiento;

import icaro.aplicaciones.informacion.dominioClases.aplicacionTalkingAgent.*;
import java.util.*;

global Set intentions;

rule "Preguntar tema" salience 100
    when
        b:Belief(code=="visitanteNuevo")
    then
        Intention intention = new Intention("preguntarTema", "gestion");
        intentions.add(intention);
end

rule "Preguntar problema" salience 100
    when
        b:Belief(code=="temaRespondido")
    then
        Intention intention = new Intention("preguntarProblema", "gestion");
        intentions.add(intention);
end

rule "Responder consulta" salience 100
    when
        b:Belief(code=="problemaRespondido")
    then
        Intention intention = new Intention("responderConsulta", "gestion");
        intentions.add(intention);
end
```

Cuadro 3.2: Ejemplo de reglas de decisión

Planificación

En esta parte, se busca una manera de llevar a cabo las intenciones generadas en el nivel anterior. Concretamente, se genera un texto que concuerde con aquello que el agente quiere comunicar. Para ello se hace uso de otro sistema de razonamiento basado en casos, en el cual se almacenan fragmentos de texto asociados a distintas intenciones del agente, a distintos contextos de la conversación y a distintos carácteres del agente, tal y como se detalla en la estructura de los casos, descrita más adelante.

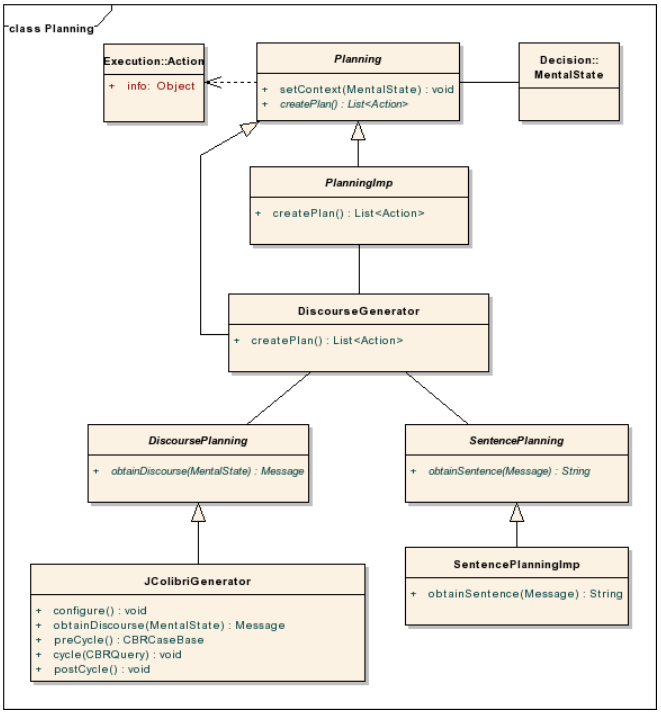

Figura 3.9: Estructura de la Planificación

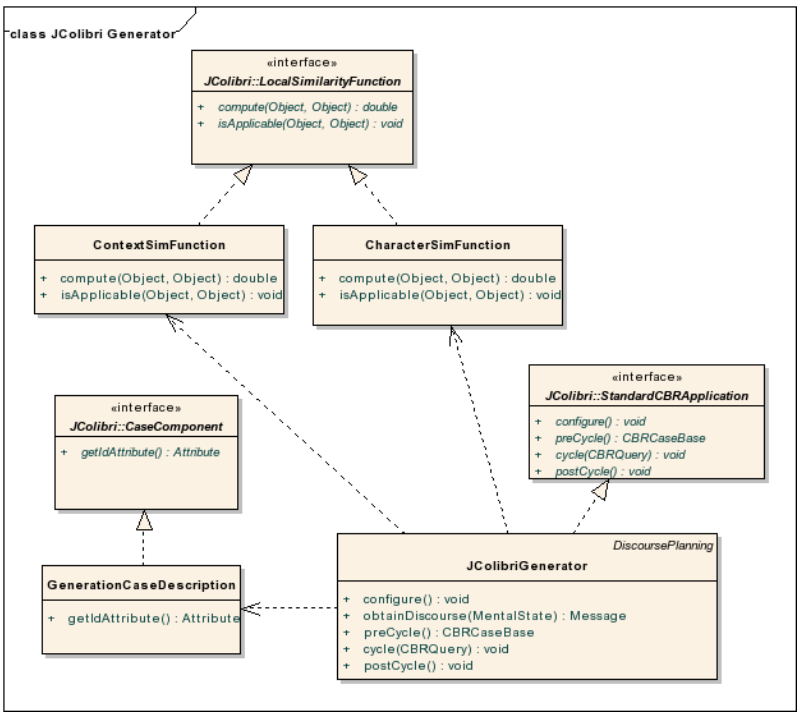

Figura 3.10: Detalle JColibriGenerator

Esta es la estructura de los casos:

Descripción del Caso

- **Intencionalidad del Agente.** es un identificador que representa la intención del agente sobre lo que debe decir. Éstas intenciones dependen de las decisiones que tome el agente. Éstas son las intencionalidades identificadas, pero se pueden añadir más:
  - darBienvenida → la intencionalidad que representa que el agente quiere dar la bienvenida
  - preguntarTema → representa que el agente quiere preguntar sobre el tema de la consulta
  - preguntarProblema → representa que el agente quiere preguntar sobre el problema concreto
  - preguntarConfirmar → representa que el agente quiere obtener una confirmación
  - preguntarMotivo → representa que el agente quiere preguntar el motivo
  - responderConsulta → representa que el agente quiere responder a la consulta realizada
  - responderConsultaDeFormaProfunda → representa que el agente quiere responder a la consulta realizada de una forma más profunda
  - rechazarCandidato → representa que el agente quiere rechazar al candidato
  - despedida → representa que el agente quiere despedirse
  - ...
- **Contexto de la conversación:** coincide con el campo de "conocimiento adquirido" del formato de casos anterior. Representa que la descripción de este caso se cumple si la conversación tiene ese contexto (i.e. el tema es amor, o el problema es celos). Este elemento es opcional, de tal modo que si no se pone, entonces es que el contexto puede ser cualquiera.
  - **tema**
  - **problema**
- **Carácter del agente:** identificador que representa el carácter del agente. Se han identificado los siguientes, pero podría haber más. Este elemento es opcional, de tal modo que si no se pone, entonces es que el carácter puede ser cualquiera.
  - **colérico**
  - **flemático**
  - **nervioso**
  - **apático**
  - ...

Solución del Caso

- **DiscursoGenerado:** texto que va a pronunciar el agente directamente cuando se produzca la situación adecuada. Este texto puede contener *referencias*, que se representan entre símbolos de @, y se explican más adelante.

Estos son unos ejemplos de casos:

Descripción	
Intencionalidad	darBienvenida
Contexto	_
Carácter	_
Solución	
Discurso Generado	Bienvenido, humano

Descripción	
Intencionalidad	responderConsulta
Contexto	tema amor
Carácter	_
Solución	
Discurso Generado	El amor es una cosa muy complicada

Descripción	
Intencionalidad	responderConsultaDeFormaProfunda
Contexto	problema celos
Carácter	colerico
Solución	
Discurso Generado	Los malditos celos son como una soga que se enrosca en tu cuello

Referencias Las referencias son un mecanismo que sirve para enriquecer los discursos generados de la base de casos de generación. Son elementos que se insertan dentro de estos discursos para, a la hora de obtenerlos, ser sustituidos por otros fragmentos de textos en distintas condiciones, incluso de forma recursiva.

Las referencias contienen los siguientes elementos:

- **Tipo de referencia:** identificador que define el tipo de ésta referencia. Sirve para poder establecer equivalencias entre referencias, de tal manera que las del mismo tipo se puedan intercambiar, sin alterar el sentido del discurso. Estos son algunos tipos posibles:
  - **refVisitante**
  - **refPropioOraculo**
  - **refConjuntoDeOraculos**

  - **refLugar**
  - **refTema**
  - **refProblema**
  - **...**

- **Contexto de la conversación:** coincide con el campo "contexto de la conversación" del formato de casos anterior. Indica que la referencia es válida si la conversación tiene ese contexto. Este elemento es opcional, y se puede indicar que la referencia es independiente del contexto.
- **Carácter del agente:** identificador que representa el carácter del agente. Los siguientes son algunos de los tipos posibles:

  - **colerico**
  - **flematico**
  - **apatico**
  - **...**

- **Texto referenciado:** es el texto por el que se va a sustituir esta referencia. Este texto a su vez puede contener otras referencias, pudiendo crear así una estructura de árbol.

Estos son unos ejemplo de referencias:

Tipo de referencia	refTema
Contexto	problema celos
Carácter	rudo
Texto referenciado	Los malditos celos, los cuales no nos afectan a nosotros, @refConjuntoDeOraculos@

Tipo de referencia	refConjuntoDeOraculos
Contexto	_
Carácter	_
Texto referenciado	los seres inmortales que todo lo conocen

Tipo de referencia	refSimil
Contexto	problema celos
Carácter	_
Texto referenciado	una soga que se enrosca en tu cuello

De este modo, en un caso de generación podría aparecer el siguiente texto:

```
@refProblema problema celos rudo@ son como @refSimil problema celos@
```

El cual, en una generación se desarrollaría formando el árbol mostrado en la figura 3.11

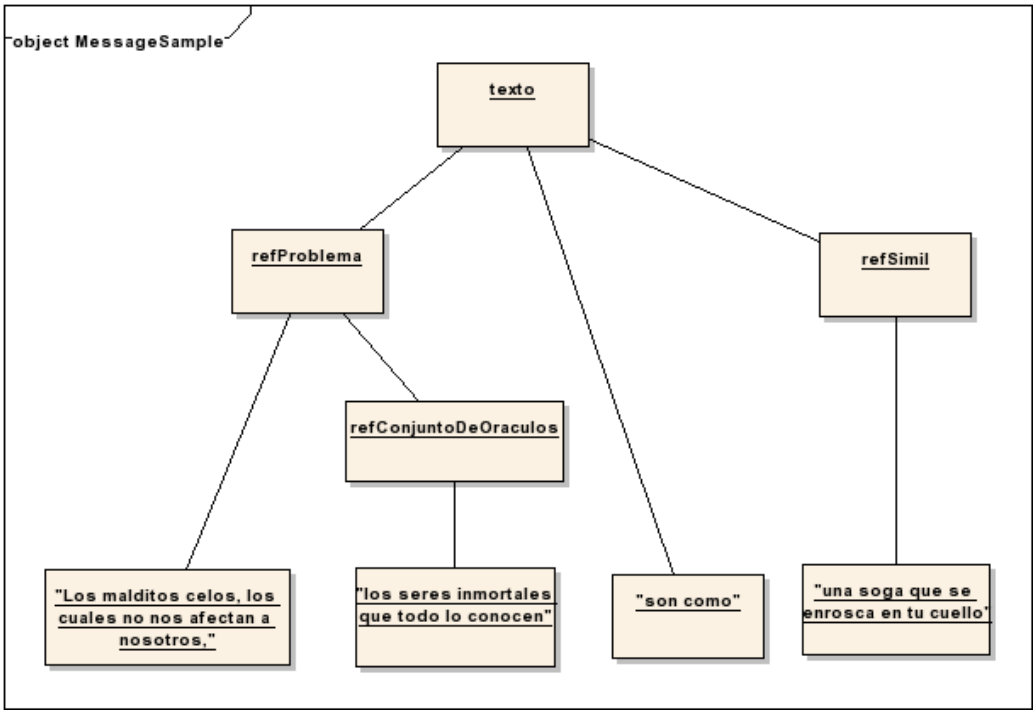

Figura 3.11: Árbol generado al resolver las referencias

La resolución finalmente daría como resultado:

```
Los malditos celos, los cuales no nos afectan a nosotros, los seres
inmortales que todo lo conocen; son como una soga que se enrosca
en tu cuello
```

Ejecución

Esta parte se encarga de llevar a cabo las acciones generadas en la fase anterior, en particular, de pasar a voz los discursos generados. Su organización se describe en la figura 3.12. Una propiedad interesante de los módulos de ejecución es que, cuando terminan su cometido, generan *hechos* en la memoria del agente. Estos hechos son creencias (belief) que se dan por seguras por el agente. Estos hechos representan la confirmación de que una determinada acción se ha llevado a cabo de manera efectiva, y pueden usarse en el proceso de razonamiento del agente.

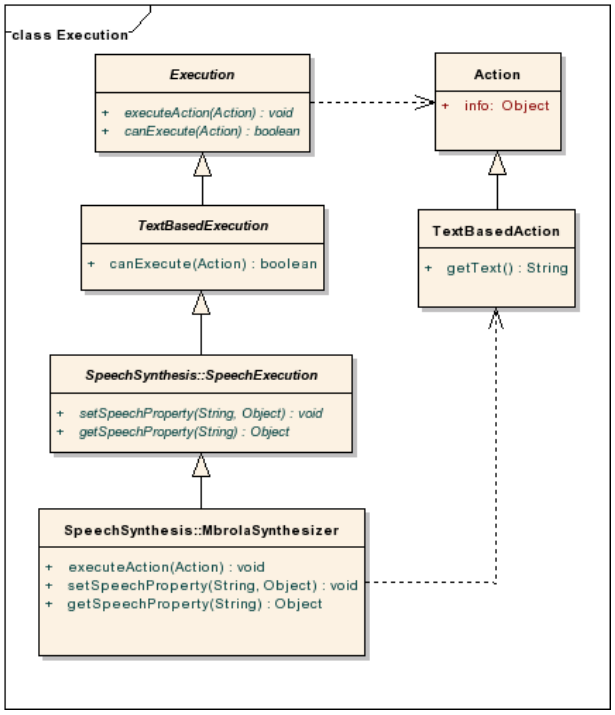

Figura 3.12: Diagrama de Clases de la Ejecución

Como puede observarse, la clase SpeechExecution representa un tipo de ejecución basada en texto que lo que hace es sintetizar el discurso en voz para emitirlo al exterior. Esta clase permite establacer ciertas propiedades de la síntesis, como la frecuencia, el tempo, el sexo de la voz utilizada, etc. Estas propiedades podrían usarse en el caso de querer representar el carácter del agente no sólo en la forma de hablar, sino en el tipo de voz. Esta misma idea se propone en [8].

3.2. Visión Dinámica

3.2.1. Visión General de la Interacción

Como en todo modelo de diálogo, se pretende establecer un flujo de información entre las dos partes implicadas en la comunicación. Este flujo de información, en su transcurso de una parte a otra, atraviesa distintos niveles de abstracción, como se muestra en la figura 3.13. Los nombres de los niveles han sido escogidos arbitrariamente.

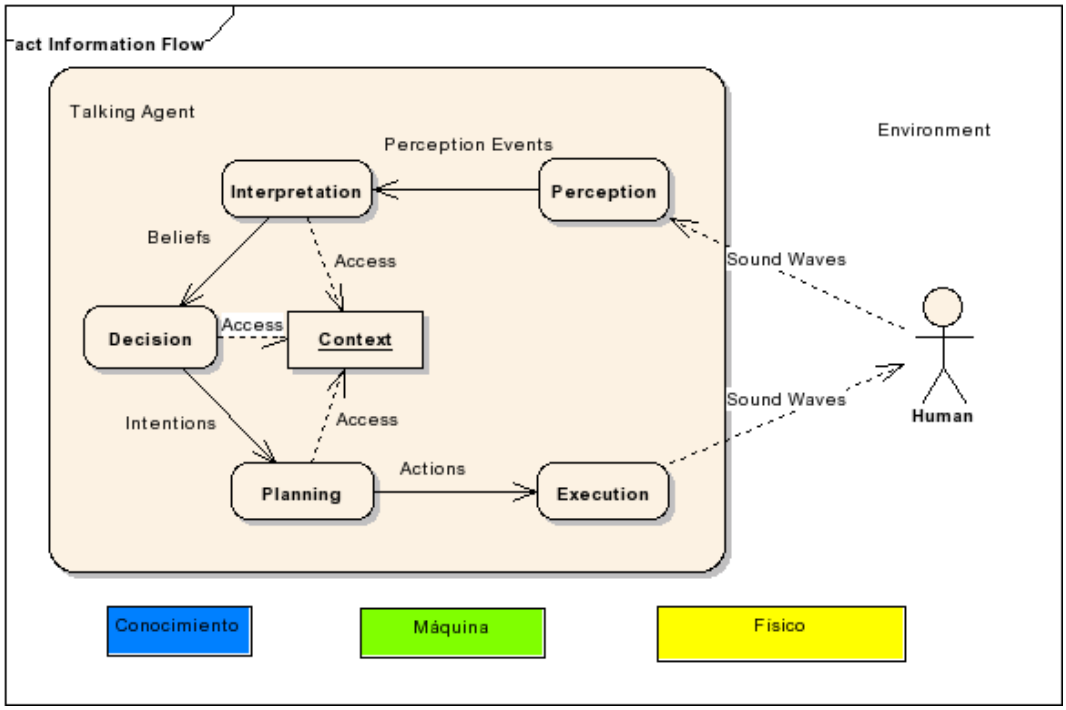

Figura 3.13: Flujo de Información

Físico. Es la información en forma física en el entorno: ondas de sonido de la voz, o señales elecromagnéticas de los sensores.

Máquina. Es la información manejada por un determinado software relativa a eventos de entrada y salida.

Conocimiento. Es información máquina condensada para representar un determinado concepto útil a la hora de tomar decisiones.

Estos niveles de abstracción se recorren en ambas direcciones, según el sentido del flujo. Cuando un agente toma la decisión de comunicar algo, genera el conocimiento en forma de intención que condensa esa decisión. Esta intención sólo dice qué hacer, pero no cómo hacerlo. Mediante un determiado módulo, se pasa a conseguir información máquina, especificando la manera de llevar a cabo la intención. Posteriormente, otro módulo la lleva a cabo de manera efectiva, produciendo la información física.

En el otro sentido, cuando un agente recibe información física, por ejemplo en forma de ondas de sonido, un módulo se encarga de transformar esta información en información máquina, manejable por el agente. Posteriormente, un módulo de interpretación se encarga de obtener el conocimiento según esa información máquina. Este conocimiento ya puede ser usado en la toma de decisiones del agente. Las interacciones entre estos módulos se muestran en la figura 3.14.

Este modelo de flujo condensa cualquier tipo de comunicación posible entre un agente y otra entidad externa, favoreciendo la comunicación multimodal. Una parte

importante del sistema se concentra en el módulo de Decision, el cual contiene la lógica que utiliza el agente para crear nuevas intenciones a partir de los hechos y creencias.

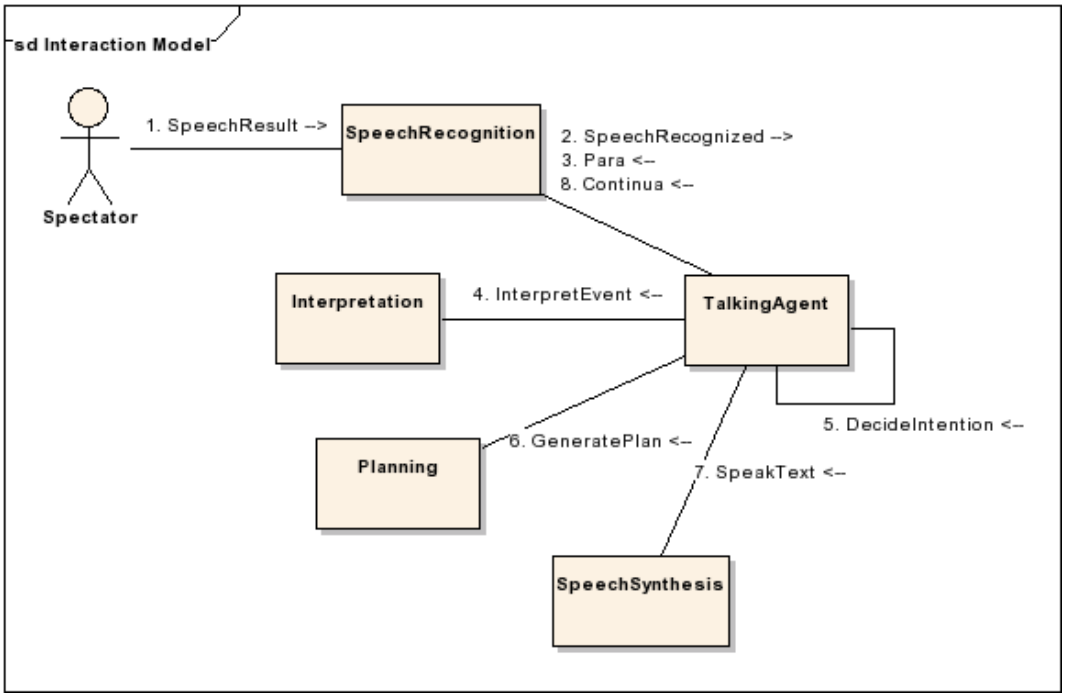

Figura 3.14: Interacción entre los diferentes módulos

3.2.2. Traza de Interacción

A continuación se detalla un ejemplo de interacción en el que un visitante entra y hace una consulta, explicando cada paso, comenzando desde el arranque del sistema.

0. **Arranque:** el framework ICARO arranca todos los agentes y recursos invocando su método *arrancar*, asignando así los correspondientes recursos del sistema a cada uno.

1. **Inicialización:** el framework manda la señal *comenzar* al agente y éste transita desde su estado inicial, siguiendo su máquina de estados, y se inicializa registrándose como suscriptor a sus recursos de percepción.

2. **Recogida de información:** el agente pregunta al resto de agentes por todo el conocimiento que ellos hayan podido adquirir en sus interacciones previas y lo añade al suyo propio. Transita al siguiente estado.

3. **Espera por visitantes:** el agente activa el recurso percepción de sensores, a la espera de la llegada de nuevos visitantes.

4. **Llegada de un visitante:** un visitante entra en la habitación, activando un sensor de movimiento. El recurso de percepción de sensores genera un evento, que envía a sus agentes suscritos. El agente transita de estado al recibir el evento y detiene el recurso de percepción de sensores.

5. **Interpretación del evento:** el agente envía el evento recibido al recurso de interpretación, el cual identifica el evento como un evento de sensores y genera un objeto "creencia" (belief) en el estado mental del agente del tipo "visitanteNuevo", haciendo transitar al agente.

6. **Decisión:** el agente invoca al sistema de razonamiento basado en reglas con su estado mental actual, el cual contiene sus creencias, y el resultado es un conjunto de objetos "intención" (intention), que indican el camino a seguir del agente. Transita al siguiente estado.

7. **Planificación de la acción:** el agente envía su estado mental, que contiene sus intenciones, al recurso de planificación, el cual comienza a generar un discurso acorde estas intenciones. Para ello hace uso del sistema de razonamiento basado en casos, utilizando la información de que dispone, y obtiene varios textos con referencias (ver Referencias 3.1.2, más arriba). Después de elegir uno al azar, comienza el proceso de resolución de referencias, en la que igualmente va escogiendo resoluciones al azar, que se ajusten a los tipos. Cuando termina lo notifica al agente y le devuelve la acción a llevar a cabo, que en este caso indica que debe pronunciarse el discurso generado. El agente transita al siguiente estado.

8. **Ejecución de la acción:** el agente envía la acción recibida al recurso de ejecución. Éste recoge el discurso contenido y lo transforma a voz. Cuando termina envía una señal al agente, el cual crea un objeto "hecho" (fact) que indica que ha hecho lo que acaba de hacer. Transita al siguiente estado.

9. **Espera por discursos:** el agente activa el recurso de percepción de voz, a la espera de un nuevo discurso.

10. **Discurso del visitante:** el visitante realiza un discurso, el cual es recogido por un dispositivo de entrada de audio conectado al recurso de percepción de voz. Éste transforma el modelo de las ondas de sonido recibidas en el texto que representan. Posteriormente genera un evento que envía a sus agentes suscritos. El agente transita de estado al recibir el evento y detiene el recurso de percepción de voz.

11. **Interpretación del evento:** el agente envía el evento recibido al recurso de interpretación, el cual identifica el evento como un evento de discurso. En este momento, se utiliza el discurso y el contexto actual del agente como descripción para buscar un caso en el sistema de razonamiento basado en casos. Tras obtener el caso con mayor puntuación, se agrega su conocimiento asociado al estado mental del agente. Cuando se termina se manda una señal al agente y este transita al siguiente estado.

12. **Decisión:** se hace lo mismo que en 6. Se supone que el agente además decide terminar la interacción.

13. **Planificación de la acción:** se hace lo mismo que en 7. Aparte, la acción generada indica que se ha de terminar la interacción.

14. **Ejecución de la acción:** se hace lo mismo que en 8. Al final se manda un evento de terminar, que hace transitar al agente al estado de compartir información.

15. **Espera por compartir información:** el agente queda a la espera de una petición de información por parte de otro agente.

16. **Compartición de información:** el agente recibe una petición para compartir información y cede su conocimiento al agente solicitante. Después de esto transita al estado final.

Capítulo 4

Arquitectura del Sistema Multiagente

A continuación se detalla la arquitectura de un sistema multiagente con varios Talking Agents, que serviría para implementar una instalación como la que se describe en el capítulo 5. Como infraestructura para el desarrollo del sistema multiagente se ha utilizado la plataforma ICARO. Previamente se da un breve repaso al mismo para posteriormente describir el modelo del sistema en sí y cómo se integran cada una de sus partes en un conjunto.

4.1. El Framework ICARO

ICARO es un framework para construir aplicaciones distribuídas, concebidas como organizaciones de dos tipos de entidades: agentes y recursos. ICARO proporciona servicios para tal organización, con el objetivo de facilitar su configuración y monitorización.

Un sistema implementado con ICARO consiste en los siguientes tipos de entidades:

Agente. Representa una entidad que puede gestionar flujos de información. Puede mandar órdenes a recursos, recibir su información y distribuírla a otros agentes y recursos. Cada agente tiene un *modelo de comportamiento*, que define la forma en que actuará en cada circunstancia. Cada modelo de comportamiento proporciona un patrón para la especificación y control del comportamiento del agente. Actualmente existe un modelo reactivo, que se basa en un autómata de estados finitos, y un modelo cognitivo, basado en la definición de un modelo de reglas y una base de conocimiento.

Recurso. Representa una entidad que lleva a cabo operaciones bajo demanda de alguna otra entidad, agente o recurso. Estas operaciones normalmente son las que satisfacen los requisitos funcionales de la aplicación.

Información. Representa una entidad que es parte del modelo de dominio de la aplicación. Las entidades de información se usan o para estrucutar infor-

mación, o para almacenar información, o para llevar a cabo cierta tarea de negocio. Son gestionadas tanto por agentes como por recursos y básicamente conforman el flujo de información del sistema.

Descripción. Representa una entidad que describe cómo los agentes y los recursos se orgnizan en el sistema, i.e. las dependencias entre ellos.

Además, ICARO proporciona facilidades de gestión, que establecen que cada agente o recurso es un *elemento gestionable*. Un elemento gestionable ofrece un interfaz para llevar a cabo operaciones de gestión en él, tales como iniciar, apagar, pausar o testear. Se usan para mantener la integridad del sistema y no tienen nada que ver con los requisitos funcionales de las aplicaciones. Por esto, las operaciones de gestión son manejadas normalmente por un conjunto de agentes predefinidos llamados *gestores*. Estos agentes tienen la responsabilidad de poner el sistema a trabajar al inicio (i.e. asignar los recursos necesarios, instanciar objetos, etc), comprobar y mantener la integridad del sistema, y posiblemente de parar el sistema.

4.2. Arquitectura

4.2.1. Visión General de la Arquitectura

Los Talking Agents, como sociedad compuesta por varias entidades autónomas, han sido diseñados como un sistema multi agente en el cual cada agente tiene la capacidad para interactuar con humanos como un agente conversacional, pero también para comunicarse con otros agentes como en un SMA típico. Esta comunicación interagente es necesaria por propósitos de gestión y coordinación. Los requisitos de gestión vienen de la naturaleza del despliegue y configuración de los agentes, dado que cada agente puede estar situado en una localicación distinta y hacer uso de recursos distribuídos. Los requisitos de coordinación se derivan de que los agentes tengan que colaborar hacia los objetivos del sistema gobal, en este caso creando una experiencia al espectador de estar inmerso en una sociedad concreta.

La arquitectura de los Talking Agent sigue el paradigma de organización de ICARO, el cual ya aborda la capacidad de distribuir los componentes. Con este paradigma, cada componente es considerado un agente si tiene un cierto grado de autonomía (para tomar decisiones), o un recurso si sólo reacciona a peticiones de servicio. Desde este enfoque, existen piezas no activas, como los módulos de Percepción, Interpretación, Planificación y Acción, cuya función se detalla en el capítulo 3, que se consideran recursos, mientras que el control y orquestación del uso de dichos recursos estará localizada en los agentes, encarnando a los Talking Agents. Además, éstos componentes conviven con los predefinidos del framework, los ya mencionados *gestores*, los cuales se encargan de las labores de gestión del sistema. La visión general del sistema es la de la figura 4.1.

Esta arquitectura distribuida tiene semejanzas con la descrita en [9], pero con ICARO ya existe una infraestructura de gestión que se ocupa de la inicialización y configuración del sistema, lo cual es normalmente un asunto complicado en los sistemas distribuidos.

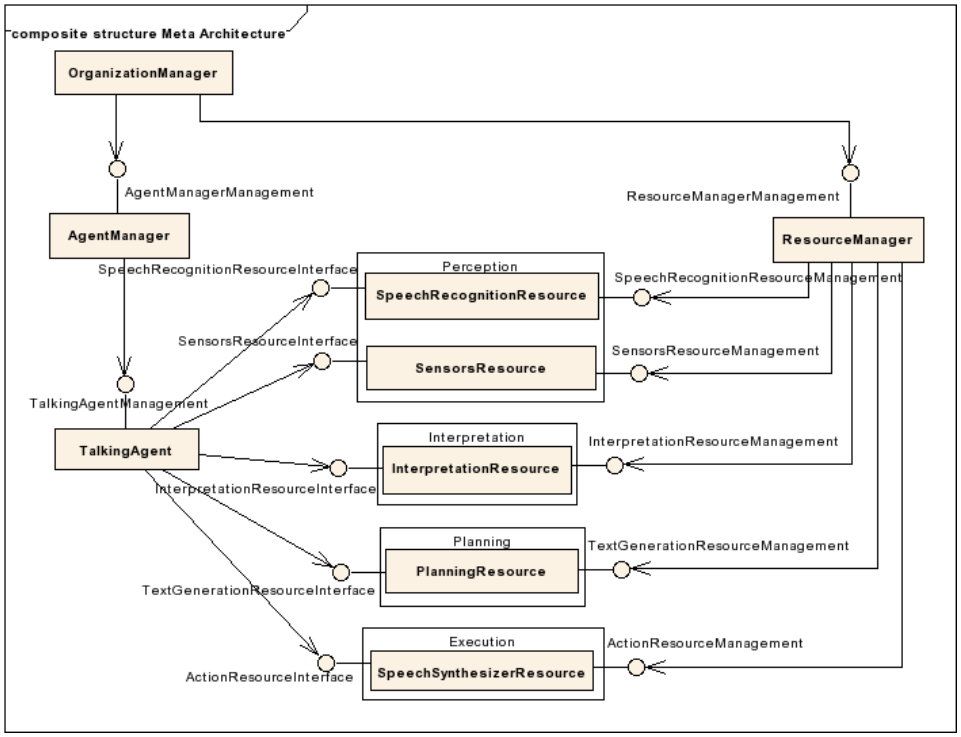

Figura 4.1: Relaciones en ICARO

La arquitectura del sistema puede dividirse en tres tipos de entidades, que se explican a continuación.

4.2.2. Agentes Gestores

Gestor de Organización. Este agente es responsable de la configuración, inicialización y monitorización del sistema completo. Puede detectar problemas en el proceso de inicialización debidos al mal funcionamiento de los componentes o de la configuración. Delega en el Gestor de Agentes para gestionar los agentes y al Gestor de Recursos para gestionar los recursos.

Gestor de Agentes. Este agente es responsable de la configuración, inicialización y monitorización de los agentes del sistema. Puede detectar problemas en este proceso y los notifica al Gestor de Organización.

Gestor de Recursos. Este agente es responsable de la configuración, inicialización y monitorización de los recursos. Puede detectar problemas en este proceso y los notifica al Gestor de Organización.

4.2.3. Agentes de Aplicación

Talking Agent. Este es el único tipo de agente de control en el sistema. Está implementado usando el patrón de agente reactivo de ICARO, de modo que su

modelo de control y comportamiento se define como una máquina de estados finitos. Puede haber varias instancias de este tipo de agente. Su implementación está encapsulada en el componente Decision, cuyas dependencias pueden observarse en la figura 4.7.

4.2.4. Recursos de la Aplicación

Recurso de Sensores. Este recurso puede procesar las señales recibidas por distintos sensores del entorno para generar los correspondientes eventos que serán enviados a los agentes subscriptores. La interfaz de uso del recurso es como muestra la figura 4.2.

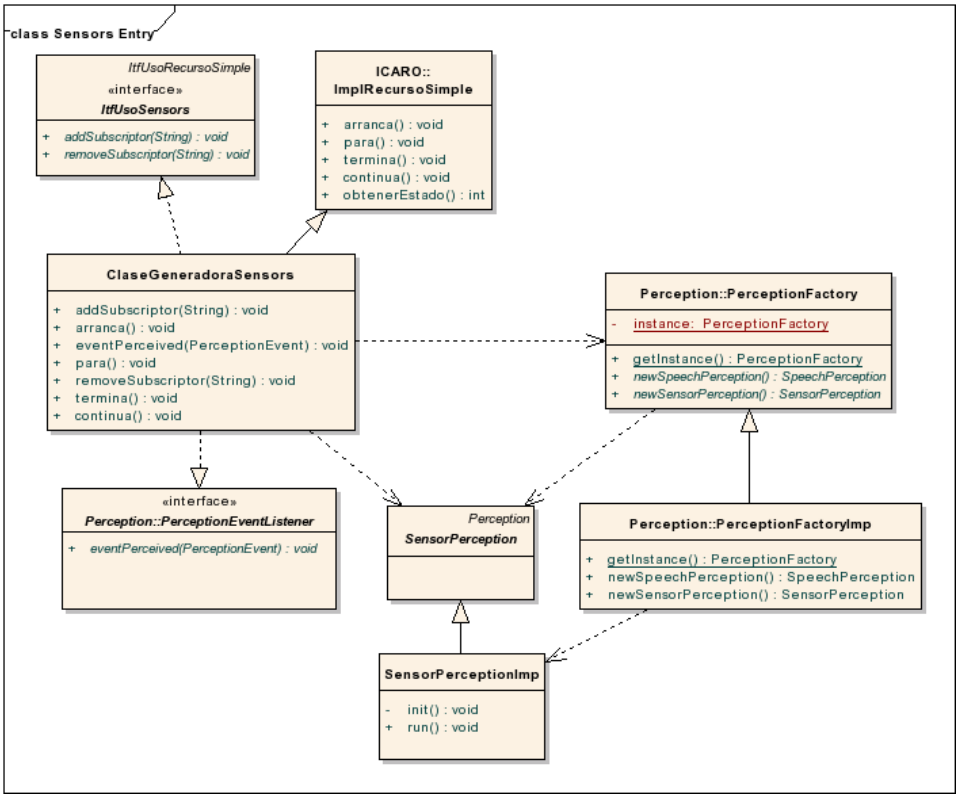

Figura 4.2: Interfaz de uso del recurso de sensores

Recurso de Reconocimiento de Voz. Este recurso es capaz de procesar las ondas de sonidos recibidas por algún dispositivo de entrada de sonido, correspondientes a cierto discurso, para reconocer las palabras pronunciadas. Entonces genera eventos que envía a los agentes subscriptores con esa información. Este componente envuelve componentes software existentes que satisfacen la Java Speech API con las interfaces requeridas para el framework ICARO (interfaces de uso y de gestión), en particular la librería MSAPI, como puede observarse en la figura 4.7. La interfaz de uso del recurso es como muestra la figura 4.3.

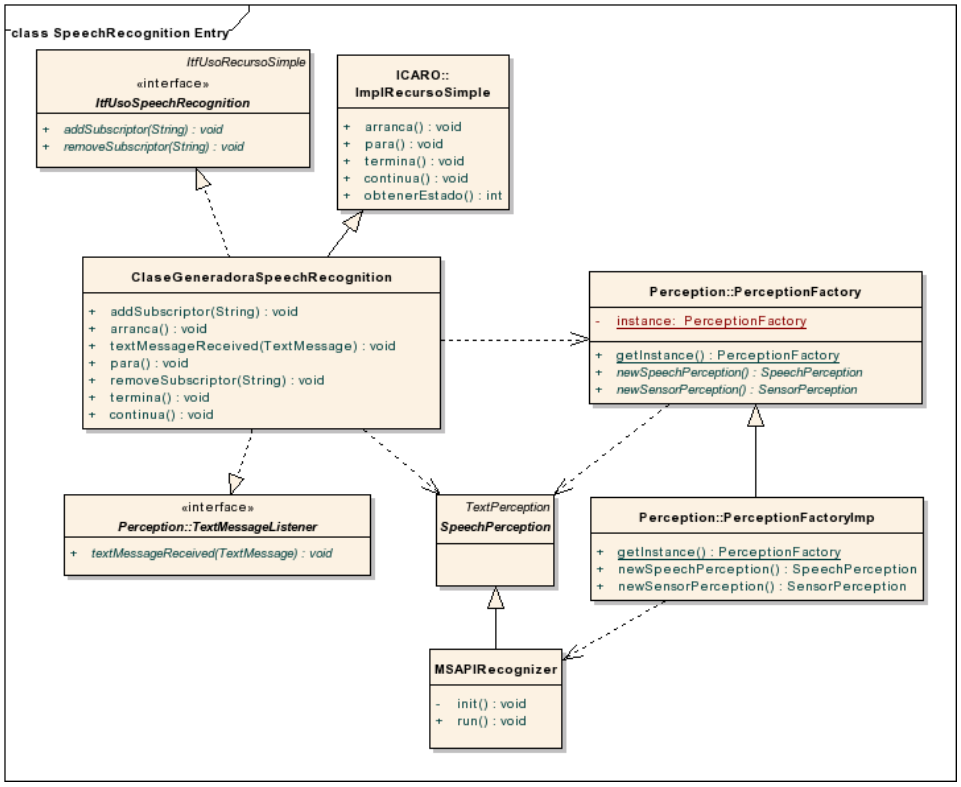

Figura 4.3: Interfaz de uso del recurso de reconocimiento de voz

Recurso de Interpretación. Este recurso consiste en distintos componentes que pueden interpretar eventos de precepción (sensores y discursos). Éstos obtienen conocimiento parcial sobre la interacción que aún debe procesarse en un nivel superior, teniendo en cuenta otra información, como interacciones previas (que podrían conformar el contexto del diálogo) o el estado actual del agente, con el objetivo de obtener la interpretación correcta de los eventos en forma de creencias. Este recurso utiliza un sistema de razonamiento basado en casos para obtener sus interpretaciones basado en JColibri, como puede observarse en la figura 4.7. La interfaz de uso del recurso es como muestra la figura 4.4.

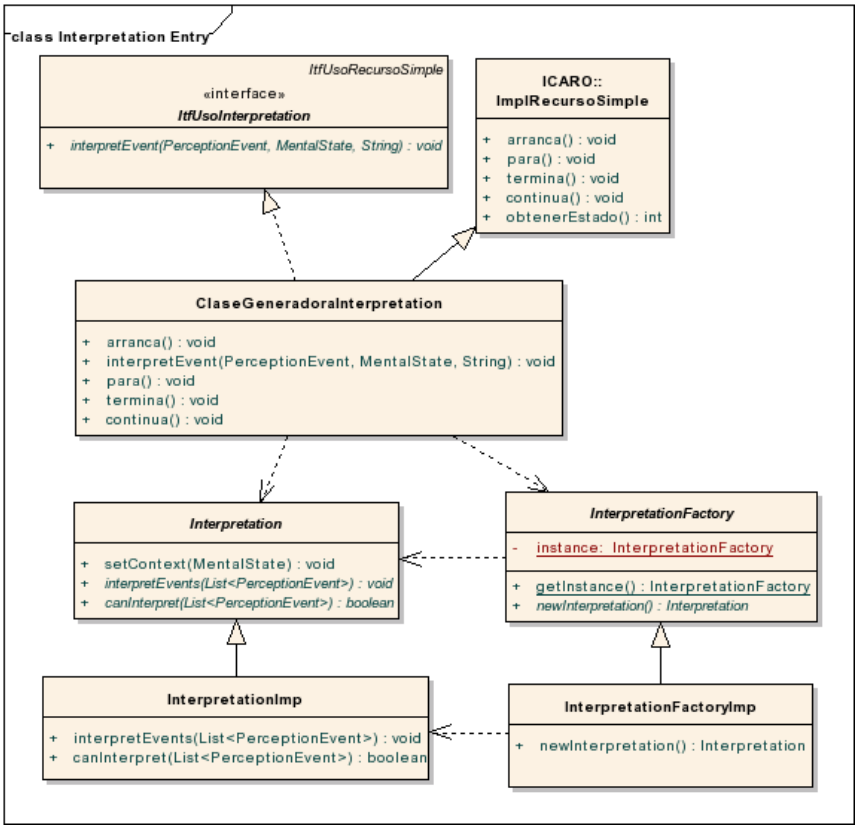

Figura 4.4: Interfaz de uso del recurso de interpretación

Recurso de Planificación. Este recurso puede determinar cómo las intenciones del agente se van a llevar a cabo, i.e. puede construir el texto para expresar la respuesta que un determinado agente quiere comunicar. Este recurso utiliza un sistema de razonamiento basado en casos para obtener el texto basado en JColibri, como puede observarse en la figura 4.7. La interfaz de uso del recurso es como muestra la figura 4.5.

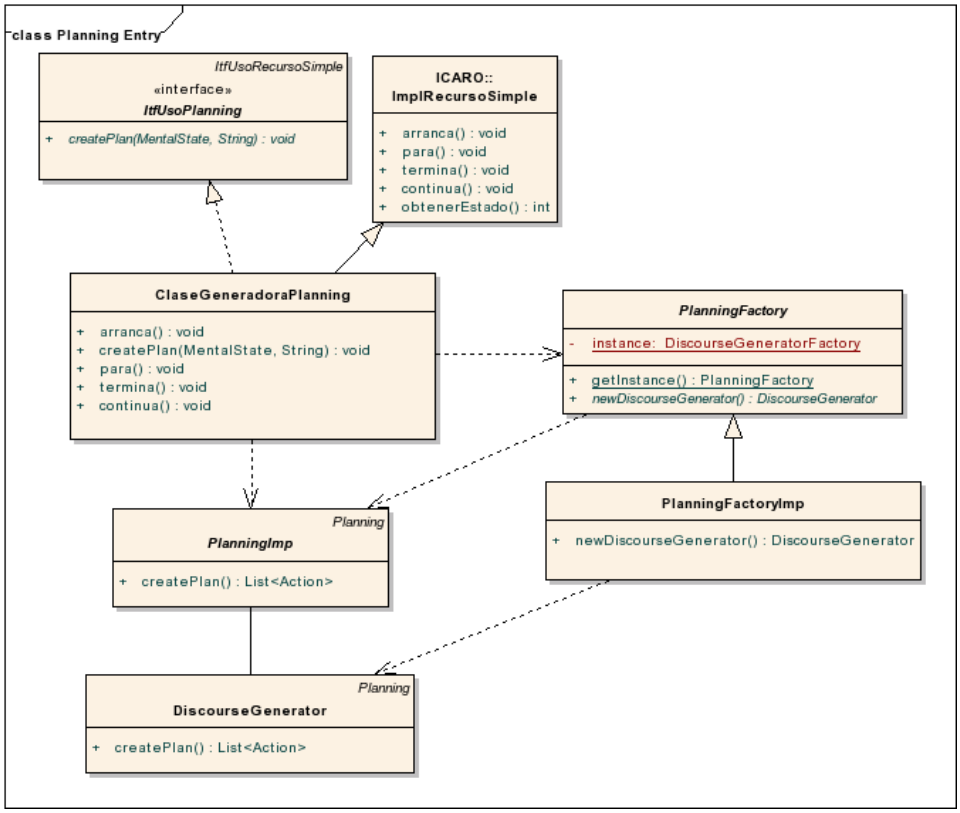

Figura 4.5: Interfaz de uso del recurso de planning

Recurso de Síntesis de Voz. Este recurso es capaz de convertir el texto proporcionado por el agente en palabras habladas que se reproducen en un cierto dispositivo de salida de sonido. Este componente envuelve componentes software existentes con los interfaces requeridos para el framework ICARO, en particular, el programa TextSound, como se muestra en la figura 4.7. La interfaz de uso del recurso es como muestra la figura 4.6.

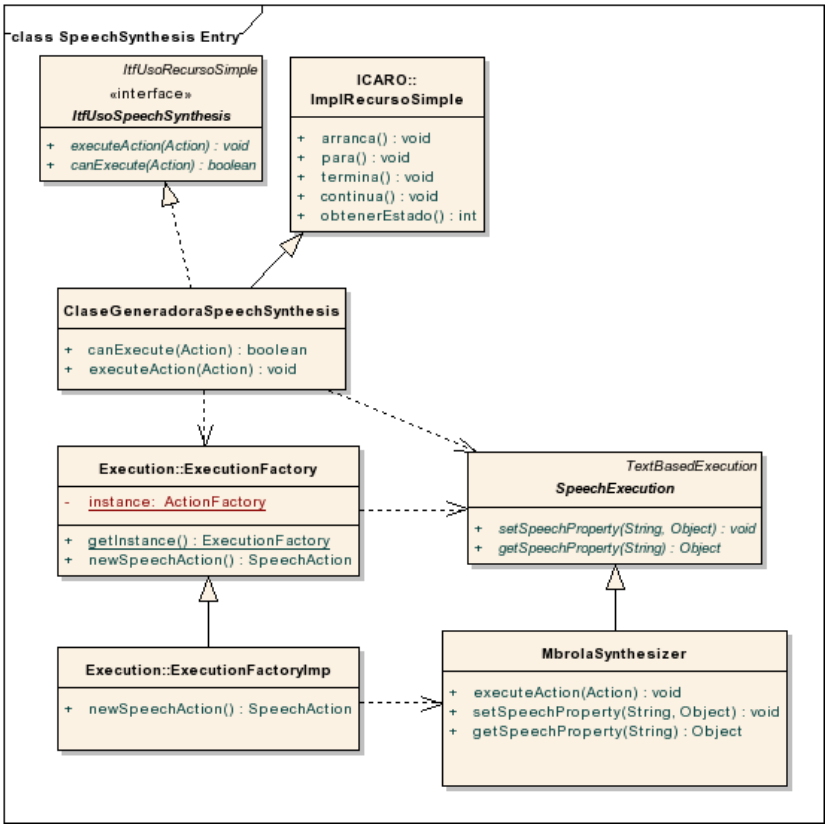

Figura 4.6: Interfaz de uso del recurso de síntesis de voz

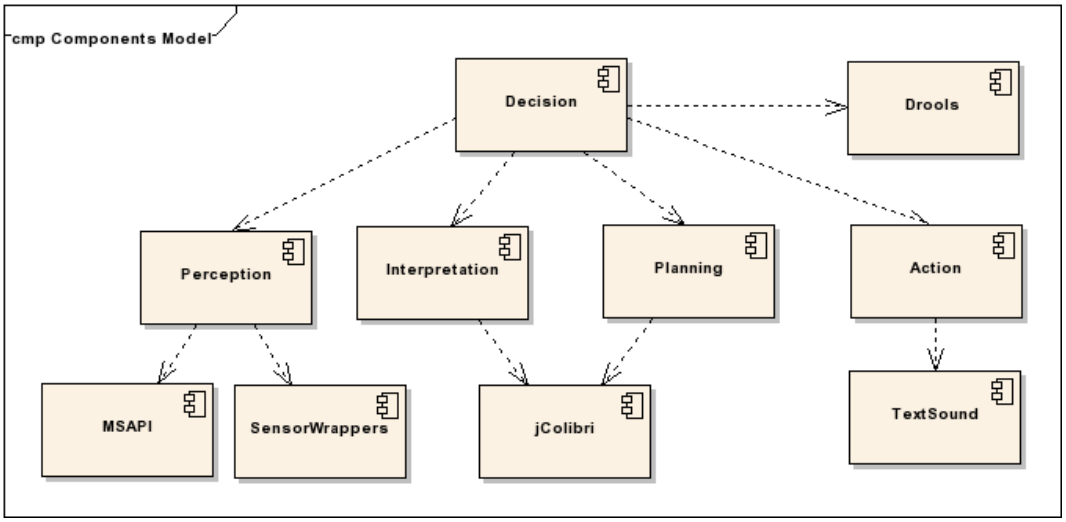

Figura 4.7: Dependencias entre componentes

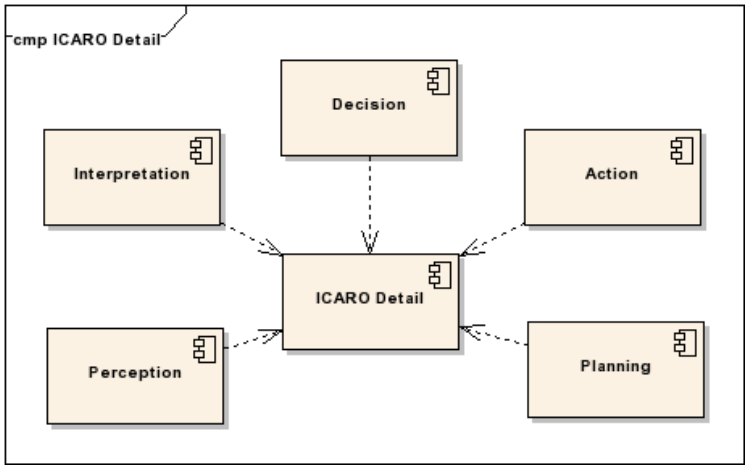

Figura 4.8: Dependencias de los componentes con ICARO

4.2.5. Ejemplo de Despliegue

Aquí se muestra un ejemplo de despliegue del sistema de forma distribuida, haciendo uso del framework ICARO, el cual utiliza la tecnología CORBA para la invocación remota de métodos. Como se puede observar, las tres instancias de Talking Agent pueden estar situadas dentro del mismo nodo, y pueden compartir los recursos de interpretación y planificación, debido a su naturaleza sin estado. Sin embargo, resulta más conveniente que cada instancia de Talking Agent haga uso de una instancia de los recursos de percepción y ejecución propia, y que éstos además estén distribuídos en distintos nodos. Esto es así porque en principio, cada representación de Talking Agent va a estar en una habitación distinta, y cada una tendra su propio micrófono y altavoz.

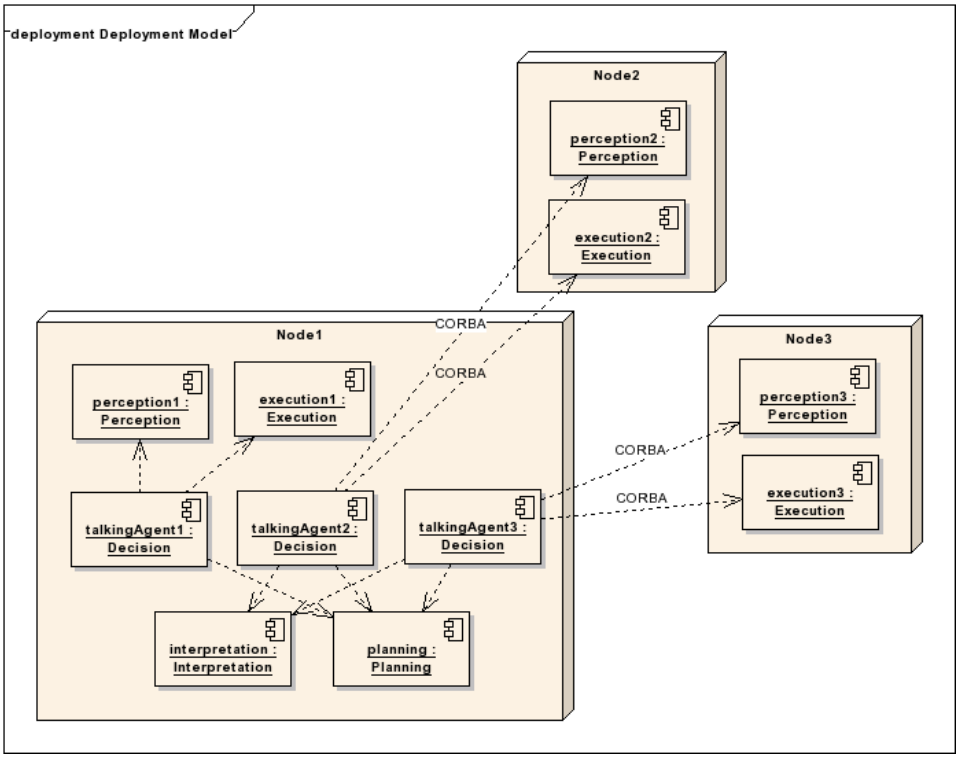

Figura 4.9: Despliegue de los componentes

Capítulo 5

Ejemplo de Uso: la Instalación Artística

Los Talking Agents se han desarrollado con el propósito de implementar instalaciones artísticas. En concreto, son la base de la instalación *Oráculos*, que se utiliza aquí para ilustrar un escenario de su aplicación y validar tanto el concepto como la implementación de Talking Agents.

Oráculos es una instalación artística interactiva que recrea las condiciones donde un oráculo recibe al espectador que desea consejo. El término "oráculo" designa tanto a la divinidad consultada, como al mediador humano que transmite la respuesta, al lugar sagrado donde se se realiza la consulta o a la respuesta dada. Cada espectador deberá enfrentarse, individualmente, a la consulta del oráculo.

La instalación Oraculos consta de tres elementos artísticos con tres doradas cabezas humanas, internamente pobladas por tres Talking Agents que reciben al espectador (figura 5.1). Cada elemento reconoce los discursos y movimientos del espectador y genera respuestas habladas. El espectador se mueve de un Talking Agent a otro navegando hacia las profundidades de su respuesta final. Además de la interacción directa con el espectador, los Talking Agents colaboran entre ellos para afinar sus adivinaciones sobre las intenciones del espectador.

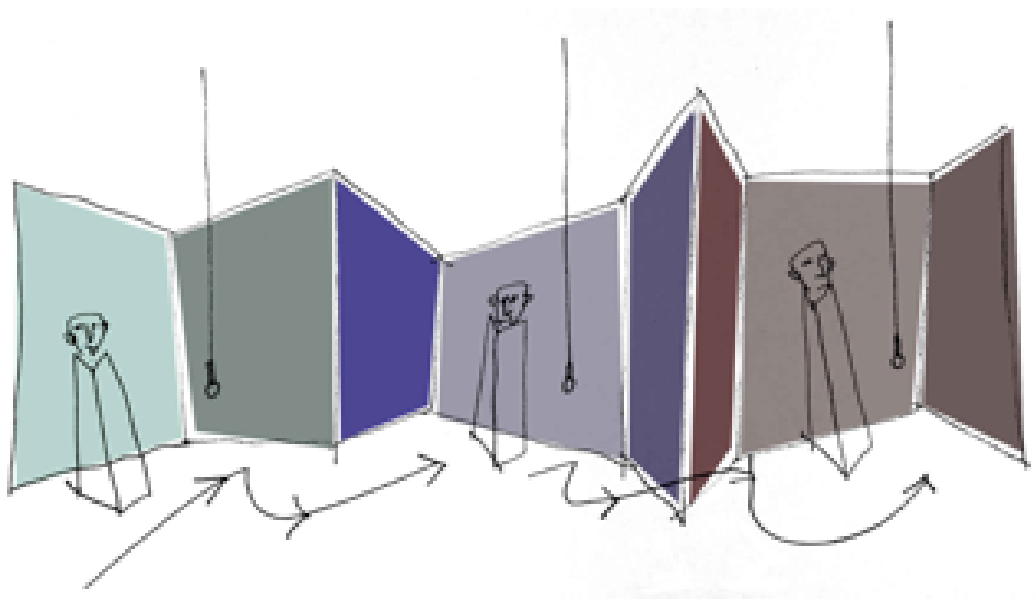

Figura 5.1: Disposición de los oráculos

La intención hacia el espectador es proveer una experiencia activa, creativa y sensitiva. Desde el punto de vista artístico, la instalación se inscribe dentro de la tradición de las representaciones teatrales interactivas, donde el papel del espectador le convierte en el sujeto y el objeto de la acción. Un juego de preguntas y respuestas donde no hay ganador, donde cada uno no tiene sólo que participar, sino actuar en el trabajo.

A continuación se muestran dos escenarios de uso de esta instalación: cuando la consulta se limita a un único agente y cuando se consultan a los múltiples agentes.

5.1. Escenarios

5.1.1. Escenario 1: un único agente

En este primer escenario se muestra un ejemplo de interacción de un espectador con un único agente. Hay que tener en cuenta que el "guión" que sigue el agente para efectuar las preguntas y dar respuestas es configurable a partir de las reglas del módulo de decisión, por tanto, el diagrama de actividades de las figuras 5.2 y 5.3 podría cambiarse.

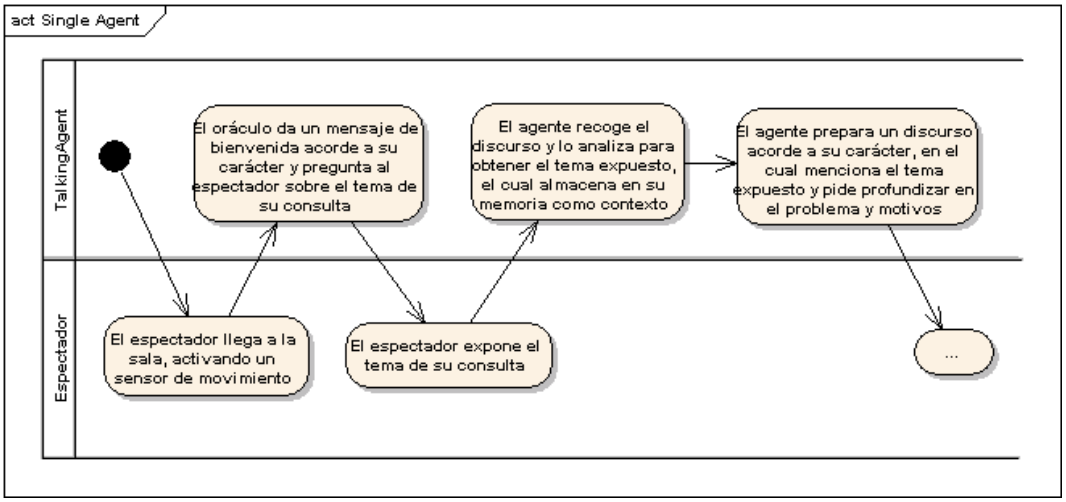

Figura 5.2: Diagrama de actividades del escenario 1 (primera parte)

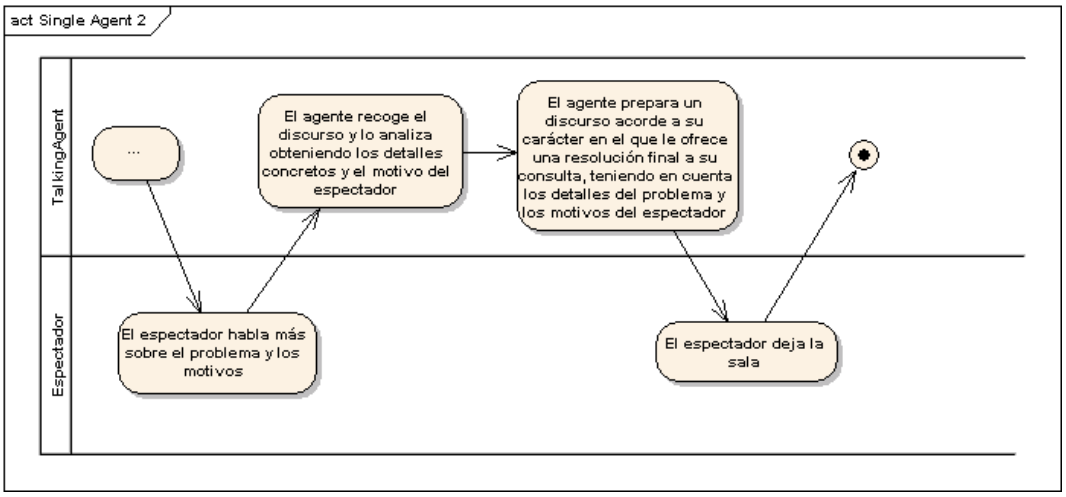

Figura 5.3: Diagrama de actividades del escenario 1 (segunda parte)

Esto es un ejemplo del escenario descrito, considerando un agente de carácter rudo.

Espectador: (Entra en la sala y activa el sensor)

Oráculo: ¡Por las barbas de Zeus! Un insignificante mortal que viene a interrumpir mi sueño. ¡Habla insecto! ¿Sobre qué quieres que este ser omnisciente te responda?

Espectador: Pues resulta que con esto de la crisis mi hijo está sin trabajo y no sabemos qué narices es lo que va a pasar.

Oráculo: El trabajo es algo que sólo los reyes pueden permitirse evitar. Pero dime, estúpido mortal, ¿por qué deseas mi consejo?

Espectador: Porque me interesa mucho lo que me tengas que decir.

Oráculo: Tus motivos no tienen ninguna relevancia para mí, pero escucha esto: en la lucha entre el mundo y tú, ponte de parte del mundo. Y ahora, desaparece de mi vista.

Espectador: (Sale de la habitación)

5.1.2. Escenario 2: múltiples agentes

En el segundo escenario, se muestra un ejemplo de interacción de un espectador con un conjunto de agentes, de manera separada. Aquí se muestra explícitamente cómo el segundo agente recibe la información recogida por el primer agente anteriormente para así poder continuar con el guión establecido.

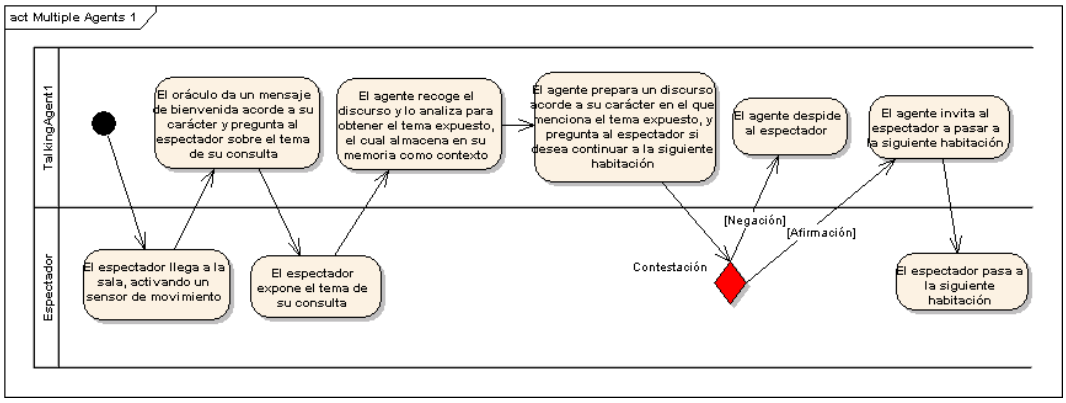

Figura 5.4: Diagrama de actividades del escenario 2. En la primera sala

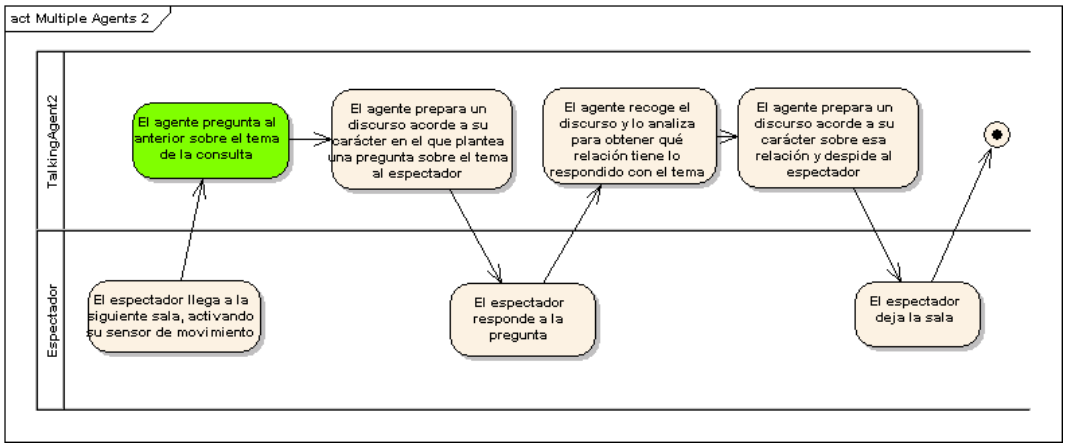

Figura 5.5: Diagrama de actividades del escenario 2. En la segunda sala

Esto es un ejemplo del escenario descrito, considerando que el primer agente es de carácter alegre y el segundo de carácter despreocupado.

Espectador: (Entra en la sala y activa el sensor)

Oráculo 1: ¡Fantástico! Un bello visitante que viene a hacerme una visita. ¡Habla bello visitante! ¿Sobre qué quieres que yo oráculo ancestral te responda?

Espectador: Ehm... pues por ejemplo sobre si el Madrid va a ganar la liga con el paquete este de Florentino, que no hace más que liarla.

Oráculo 1: El deporte es el opio de los pueblos. Aún no estás preparado para afrontar nuestra verdad, ¿deseas, gracioso humano, pasar a la siguiente habitación?

Espectador: Sí, claro.

Oráculo 1: Avanza, bello visitante.

Espectador: (Pasa a la siguiente habitación y activa el segundo sensor)

Oráculo 2: (Después de obtener la información del anterior oráculo) ¡Oh vaya, un humano! ¡Bah! Dime, humano ¿qué tiene ese deporte, sea cual sea, para gustaros tanto?

Espectador: Pues yo que sé, es emocionante.

Oráculo 2: Vosotros siempre os regís por vuestros aburridos sentimientos. Atento mortal: cuando decimos que la catástrofe era inevitable, nos quedamos tranquilos un tiempo, aunque sólo por poco tiempo. Y ahora, devuélveme mi soledad.

Espectador: (Sale de la habitación)

5.2. Discusión de los Resultados

La impresión general de las conversaciones es bastante buena, ya que los diálogos que se obtienen resultan bastante naturales. Sin embargo, hay que tener en cuenta que gran parte del funcionamiento de estas conversaciones depende del hecho de que el agente en todo momento lleva la iniciativa de la conversación, decidiendo en cada caso quién debe responder o preguntar y qué. Por esta razón, en su razonamiento se incluyen las espectativas de respuesta del usuario, las cuales indican en gran medida qué es lo que debe entender el agente de lo que le dice el usuario.

En lo referente al *análisis de lenguaje*, se hacen los siguientes apuntes:

En el escenario 1, cuando el agente pregunta al usuario qué quiere que le responda, inmediatamente espera una respuesta donde el usuario mencione el tema de su consulta. Así, cuando recibe su discurso, busca en el cbr un caso que reúna la descripción de la respuesta esperada y las palabras contenidas en el discurso, encontrando un caso que identifica las palabras ”trabajo” y ”crisis”, el cual determina que el tema de la consulta es el trabajo. Entonces lo menciona en su siguiente discurso, antes de preguntar de nuevo.

En la nueva respuesta del usuario, el agente espera encontrar el motivo del espectador, dado que es lo que se le ha preguntado antes. Sin embargo, al buscar en el cbr no encuentra un caso con las palabras adecuadas (porque el usuario no ha dicho nada concreto), lo cual se demuestra en que el agente se ve obligado en su siguiente discurso a utilizar una referencia ”comodín” al motivo explicado (”tus motivos no tienen relevancia para mí”).

En el escenario 2, la primera pregunta se resuelve igual que en el escenario 1, identificando la palabra ”liga” en un caso dentro de la base de casos, que indica que el tema es el deporte, con lo que se menciona en el siguiente discurso del agente. En ese momento se hace una nueva pregunta, en la que la respuesta esperada es una confirmación.

Las confirmaciones se tratan igual que cualquier otro tipo de respuesta, teniendo en cuenta que el conocimiento que se adquiere de éstas es del tipo ”afirmación” o ”negación”. En este caso se reconocen las palabras ”sí” y ”claro” en un caso que indica que la respuesta es afirmativa.

Por último, la palabra ”emocionante” indica al agente que el motivo tiene que ver con los sentimientos, con lo que incluye esa referencia en su discurso posterior.

En lo referente a la *síntesis de lenguaje*, se hacen los siguientes apuntes:

Se puede observar que las fórmulas de saludo inicial en los dos escenarios siguen una misma estructura, modificada por el carácter del agente. Esto es debido a que sólo existe un único caso en la base de casos para esa situación, que define la estructura del discurso de esta manera, en la que existen partes variables en forma de referencias, que se sustituyen de una forma u otra según el carácter del agente o el contexto de la conversación. Puede verse por ejemplo, que en algunas situaciones las referencias se repiten. Esto es porque no se han definido más del tipo adecuado.

Luego existen diversas fórmulas para mencionar los temas o motivos identificados en la conversación, incluyendo fórmulas ”comodín” para cuando éstos no han conseguido ser determinados. Al final del escenario 1, se hace una mención al tema de la conversación (”en la lucha entre el mundo y tú, ponte de parte del mundo”) que aunque parezca algo críptica, es correcta. Al fin y al cabo se trata de oráculos y se pretende que sus respuestas no sean del todo claras, pero que estén relacionadas con el tema que se está tratando. Lo mismo ocurre al final del escenario 2.

Finalmente, se llega a la conclusión de que tanto la riqueza en el entendimiento del lenguaje como en la generación, depende en gran medida del volumen de las bases de casos relacionadas, lo cual es en principio una desventaja, porque hace falta un cuerpo inicial para comenzar a funcionar, pero luego se puede convertir en una ventaja, por la posibilidad de agregar más riqueza de forma sencilla, introduciendo nuevos casos.

Capítulo 6

Conclusiones

6.1. Discusión sobre el Trabajo

Este trabajo presenta la especificación de un sistema multiagente de agentes conversacionales, la cual ha sido orientada para facilitar su evolución y mantenimiento, así como su flexibilidad a la hora de procesar el lenguaje natural, teniendo en cuenta la gran imprecisión de los posibles discursos realizados por los usuarios, y la arbitrariedad de los posibles discursos generados por los agentes.

Para ello se ha optado por una arquitectura en la que se separan las cinco funciones principales de la gestión de diálogo multimodal: percepción, interpretación, control y decisión, planificación y ejecución. Lo interesante de esta arquitectura, es que cada una de estas funciones, a excepción de la de control y decisión, se hace corresponder con un *recurso* dentro del sistema multiagente. Así, cada recurso es automáticamente gestionado a bajo nivel por el framework utilizado (monitorización, arranque y parada principalmente), facilitando al desarrollador la gestión de las tareas de mantenimiento. Además, las interfaces para cada recurso están unificadas, de modo que la actualización de los recursos puede llevarse a cabo sin alterar el funcionamientdo del sistema y pueden introducirse por ejemplo mecanismos de páginas amarillas para agregar de manera transparente nuevos tipos de percepción o ejecución. Otra ventaja es que los recursos pueden distribuirse, de tal manera que puedan ser usados de manera remota, e incluso una misma instancia de recurso puede ser compartida por varios clientes en distintos nodos.

En cuanto a la función de control y decisión, se hace corresponder con un *agente* dentro del sistema. Al igual que con los recursos, esto permite delegar en el framework ciertas tareas de gestión de bajo nivel, pero lo más importante es que el modelo de comportamiento se define de manera separada, pudiendo modificarlo fácilmente sin alterar el sistema. Además, se ha construido por encima otro modelo de decisión que utiliza un sistema de razonamiento basado en reglas. De este modo, es posible modificar los criterios de decisión del agente tan sólo modificando las reglas.

Con respecto al procesamiento del lenguaje natural, se ha apostado por un enfoque basado en sistemas de razonamiento basados en casos. Sobre todo en la parte de análisis de lenguaje, difiere de los enfoques basados en parsing estadístico

explicados en la sección 2.2 en que es menos capaz de extraer información, ya que no realiza análisis sintáctico, resolución de referencias, etc; pero sin embargo afronta de una manera aceptable la arbitrariedad e imprecisión de los discursos analizados, que es uno de los objetivos de este trabajo.

6.2. Alternativas y Trabajo Futuro

A continuación se enumeran una serie de cuestiones a tener en cuenta como trabajo futuro o como posibles alternativas de implementación

Detección de emociones en los discursos. Agregar detección de emociones a la interpretación de los discursos de los usuarios. Esto añadiría una fuente más de información utilizable por el agente (p.e. responder de manera distinta según las emociones detectadas). En [8] se describe una herramienta que lleva a cabo esta tarea.

Expresión de emociones mediante cambios en la voz Agregar cambios en la voz del agente según su carácter o lo que diga en cada momento, como manera adicional de enriquecer estéticamente los discursos pronunciados. También en [8] se habla sobre esta idea.

Aprendizaje automático de los agentes Agregar algún mecanismo de aprendizaje automático a los agentes, que le permita reconocer cuándo ha errado en la extracción de información de los discursos de los usuarios y mejorar al respecto. Utilizando el sistema de razonamiento basado en casos, esto se reduce a encontrar una manera de agregar nuevos elementos a la base de casos, modificar los existentes o incluso eliminar casos erróneos, utilizando un algún mecanismo que permita evaluar la validez de una solución.

Uso de mecanismo de páginas amarillas Hacer uso del mecanismo de páginas amarillas del framework multiagente para permitir a los agentes adquirir nuevas capacidades de manera dinámica tan sólo agregando los nuevos recursos a las páginas amarillas, para poder ser detectados.

Bibliografía

[1] James Allen. Review of "natural language understanding". *Comput. Linguist.*, 14(4):96–97, 1988. Reviewer-Allen, James.

[2] James Allen, Donna Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. An architecture for a generic dialogue shell, 2000.

[3] James Allen, George Ferguson, and Amanda Stent. An architecture for more realistic conversational systems. In *In Proceedings of Intelligent User Interfaces 2001 (IUI-01*, pages 1–8, 2001.

[4] Lynne Cahill. Lexicalisation in applied NLG systems. Technical Report ITRI-99-04, ITRI, University of Brighton, 1998. obtainable at http://www.itri.brighton.ac.uk/projects/rags/.

[5] Justine Cassell. Embodied conversational agents: Representation and intelligence in user interface.

[6] Belén Díaz-Agudo, Pedro A. González-Calero, Juan A. Recio-García, and Antonio Sánchez-Ruiz-Granados. Building cbr systems with jcolibri. *Special Issue on Experimental Software and Toolkits of the Journal Science of Computer Programming*, 69(1-3):68–75, 2007.

[7] W. Eckert, T. Kuhn, H.Ñiemann, S. Rieck, A. Scheuer, and E. G. Schukat-talamazzini. A spoken dialogue system for german intercity train timetable inquiries. In *In Proc. European Conf. on Speech Technology*, pages 1871–1874, 1993.

[8] Virginia Francisco. *Identificación Automática del Contenido Afectivo de un Texto y su Papel en la Presentación de Información*. PhD thesis, Universidad Complutense de Madrid, Madrid, 11/2008 2008.

[9] J. Gustafson. *Developing multimodal spoken dialogue systems. Empirical studies of spoken human-computer interaction*. PhD thesis, KTH, Department of Speech, Music and Hearing, KTH, Stockholm, 2002.

[10] Drools Home. http://www.jboss.org/drools/.

[11] FreeLing Home. http://garraf.epsevg.upc.es/freeling/.

[12] FreeTTS Home. http://freetts.sourceforge.net/docs/index.php.

[13] Java Speech API Home. http://java.sun.com/products/java-media/speech/.

[14] Loquendo Home. http://www.loquendo.com/es/.

[15] Microsoft Speech Home. http://www.microsoft.com/speech/speech2007/default.mspx.

[16] Sphinx-4 Home. http://cmusphinx.sourceforge.net/sphinx4/.

[17] Statistical Language Modeling Toolkit Home. http://www.speech.cs.cmu.edu/SLM/toolkit.html.

[18] Talking Java Home. http://www.cloudgarden.com/JSAPI/index.html.

[19] TextSound 2.0 Home. http://www.bytecool.com/textsnd.htm.

[20] The Stanford Parser Home. http://nlp.stanford.edu/software/lex-parser.shtml.

[21] Mike Reape and Chris Mellish. Just what is aggregation anyway?

[22] Ehud Reiter and Robert Dale. A fast algorithm for the generation of referring expressions. In *Proceedings of the 14th conference on Computational linguistics*, pages 232–238, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[23] Ehud Reiter and Robert Dale. Building applied natural language generation systems, 1997.

[24] Oliver Lemon School and Oliver Lemon. Managing dialogue interaction: A multi-layered approach. In *In Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*, pages 168–177, 2003.

[25] ICARO Project Web Site. http://icaro.morfeo-project.org/.