

Including Conversational Agents into Structured Hybrid 3D Virtual Environments

P. Almajano, M. L. Sánchez, I. Rodríguez and E. Mayas

Abstract— Structured Hybrid 3D Virtual Environments are 3D virtual spaces where staff (organisational) software agents support human users in their task achievement. These systems are characterized by: i) being hybrid, so that humans and software agents can interact; and ii) being structured and task oriented, so that interactions are regulated by a subjacent Organisation Centered Multi Agent System (OCMAS) —an Electronic Institution (EI). The contribution of this paper is to include task-oriented conversational staff bots (i.e. the embodiment of staff agents in the 3D environment) that communicate with users by using natural language. With this aim, we extend the Artificial Intelligence Mark-up Language (AIML) with special tags to enable complex task-oriented conversations whose flow needs to consider both the states of the conversation and the ontology related to the task. We evaluate the usability of our conversational proposal and compare it to a previous command-based interaction system. Results show the conversational approach presents a higher user satisfaction than the command-based one. Moreover, in average, it also performs better in terms of efficiency, effectiveness and errors.

Keywords— 3D Virtual Worlds, Human-Computer Interaction, AIML

I. INTRODUCCIÓN

LOS MUNDOS virtuales 3D (3D VW) son entornos virtuales (VE) persistentes inicialmente concebidos como espacios en los que las personas interactúan con el único propósito de la socialización y el entretenimiento. Sin embargo, el actual Internet interactivo 3D tiene como objetivo involucrar a la gente en experiencias digitales 3D no sólo con estos propósitos lúdicos, sino también con otros objetivos más serios, como la educación, las compras o el diseño colaborativo de productos.

Los Entornos Virtuales 3D, Híbridos y Estructurados permiten la interacción entre múltiples usuarios en línea y programas de ordenador (es decir, agentes software) dentro de actividades serias. Sobre la base de una combinación de un enfoque de los Sistemas Multi-Agente Centrados en la Organización (OCMAS) [4] y una interfaz 3D, estos sistemas permiten a los humanos realizar tareas complejas específicas, proporcionándoles tanto una inmersión en los entornos virtuales 3D como la regulación (es decir, la estructuración) de su interacción con otros participantes.

Específicamente, utilizamos Instituciones Virtuales (VI) como Entornos Virtuales 3D, Híbridos y Estructurados que

combinan una Institución Electrónica [3] (EI, una infraestructura OCMAS) y los mundos virtuales 3D. Brevemente, las EI modelan organizaciones que estructuran la interacción de los participantes mediante la definición de protocolos de comunicación que deberán ser seguidos al realizar tareas dentro de las actividades específicas del sistema. Estos protocolos se basan en la Teoría de los actos de habla [10], de forma que ilocuciones verbalizadas cuentan como acciones. Cuando se ejecuta una institución virtual (VI), además de la especificación de la organización, la institución electrónica (EI) hace un seguimiento de su estado actual, que incluye, por ejemplo, el número actual de participantes. Aunque los participantes de la EI pueden jugar diferentes roles, este documento se centra en (i) los participantes que son agentes de software internos que juegan roles institucionales para dar soporte a las tareas de la organización, y (ii) los usuarios humanos que juegan roles externos y se unen a la organización para llevar a cabo dichas tareas.

En cuanto a los mundos virtuales 3D (3D VW), éstos proporcionan escenarios inmersivos donde los usuarios participan y siguen intuitivamente las actividades en curso. Un usuario controla un avatar en el VW e interactúa con otros usuarios y bots internos (la representación corpórea de los agentes internos) a fin de lograr sus metas. De esta manera, el estilo de interacción del usuario-agente resulta clave en la realización de las tareas. Los VW ofrecen ventanas de chat para realizar interacciones usuario-agente textuales. Un trabajo previo en un entorno virtual 3D orientado a tareas [1] implementó un estilo de interacción agente-usuario basado en comandos dentro de dichas ventanas de chat. Sin embargo, la interacción basada en comandos es propensa a errores, especialmente para los usuarios que no están familiarizados con los sistemas basados en comandos.

Las interacciones en lenguaje natural constituyen una alternativa adecuada a los sistemas basados en comandos. Los agentes conversacionales encarnados son prototípicos de este tipo de interacciones. Éstos son personajes virtuales que son capaces de entablar una conversación con humanos. Hoy en día, se pueden encontrar principalmente como asistentes virtuales que proporcionan información a los usuarios en entornos web. Específicamente, los chat-bots basados en el Lenguaje de Inteligencia Artificial de Mercado (AIML) [12] son bots reactivos ampliamente conocidos que siguen estructuras de diálogo básicas definidas en archivos estáticos. Éstos han sido concebidos para dar, bajo petición, información general a los usuarios. Además, son capaces de pedir al usuario información general (por ejemplo, sobre su nombre o género) y almacenar su respuesta en una memoria no tipada (es decir, en una lista de valores de tipo cadena).

P. Almajano, Institut d'Investigació en Intel·ligència Artificial (IIIA – CSIC), Universitat de Barcelona, España, palmajano@iia.csic.es
I. Rodríguez, Universitat de Barcelona, España, inma@maia.ub.es
M. L. Sánchez, Universitat de Barcelona, España, maite_lopez@ub.edu
E. Mayas, Universitat de Barcelona, España, enric.mayas@gmail.com

La contribución de este trabajo es la adición de un mecanismo de interacción conversacional a los Entornos Virtuales 3D, Híbridos y Estructurados. En particular, los agentes internos, encarnados en bots internos, interactúan con los seres humanos mediante un mecanismo de interacción que incorpora un nuevo sistema de conversación al sistema anterior basado en comandos. En este último, el bot interno únicamente entiende los mensajes del usuario en lenguaje de comandos. En el primero, el bot interno conversa con los usuarios en lenguaje natural y es capaz de gestionar conversaciones orientadas a tareas con las funcionalidades mencionadas anteriormente. Para ello, se extiende el Lenguaje de Inteligencia Artificial de Mercado (AIML) con etiquetas especiales que habilitan nuestras conversaciones orientadas a tareas. A la extensión resultante la denominamos AIML Orientado a tareas. Éste está creado en base a la especificación de la EI y su flujo es controlado considerando el estado actual de la EI. El mecanismo de interacción se evalúa en términos de eficacia y eficiencia del usuario, sus errores y satisfacción durante la realización de tareas estructuradas.

Este trabajo se estructura de la siguiente manera. Las siguientes secciones presentan el trabajo relacionado (Sec. 2), nuestro 3D VE conversacional y estructurado (Sec. 3) y las conversaciones orientadas a tareas (Sec. 4). Posteriormente, mostramos en la Secc. 5 los resultados de la evaluación en el contexto de v-mWater, un mercado virtual para el comercio de derechos de agua; y su comparación con el sistema anterior de comandos. Por último, concluimos con algunas observaciones y propuestas de trabajo futuro.

II. TRABAJO RELACIONADO

En la literatura podemos encontrar diversos sistemas conversacionales. De entre éstos podemos destacar los sistemas de tutoría, que son los tutores virtuales que siguen principios pedagógicos [8, 6]. Por un lado, el objetivo de AutoTutor [6] es satisfacer, en un orden específico, las expectativas de los estudiantes relacionadas con la física newtoniana. El tutor virtual es un personaje en 3D con una interfaz textual que almacena su conocimiento en un script de currículo y utiliza análisis semántico latente como algoritmo de reconocimiento de patrones. Los autores evalúan empíricamente los logros de aprendizaje de los estudiantes y la calidad de los diálogos (mediante una variación del test de Turing). Por otra parte, las conversaciones en nuestro sistema siguen protocolos complejos y necesitan considerar tanto la especificación del sistema como su estado actual. Además, nuestro objetivo es dar soporte a los usuarios para lograr sus objetivos en el sistema, en lugar de simular a los humanos, y por lo tanto, se evalúa la usabilidad de nuestro sistema en lugar de llevar a cabo un test de Turing.

Por otro lado, CHARLIE [8] utiliza un mecanismo AIML (Lenguaje de Inteligencia Artificial de Mercado) para mantener una conversación general con los estudiantes. La interfaz de usuario del bot es una ventana emergente con un área de texto dedicada a la conversación. Específicamente,

determinadas palabras clave vinculan plantillas AIML con una tarea de examen, donde los estudiantes pueden pedir exámenes predefinidos/personalizados. Del mismo modo, nuestro robot está basado AIML, pero hemos ampliado el lenguaje para dar soporte a las conversaciones orientadas a tareas. Por otra parte, nuestra interfaz de usuario es un mundo virtual 3D que simula entornos reales y proporciona una experiencia de inmersión para los usuarios humanos.

Otras interfaces textuales extienden AIML con motores de reglas [5, 9, 13]. En primer lugar, VISTA [13] (Agente Virtual Interactivo contador de historias) utiliza una interfaz web basada en XML para responder las preguntas de los usuarios acerca de las historias. Las consultas no cubiertas por patrones AIML son procesadas por un motor basado en lógica a través de una base de conocimientos dinámica. En segundo lugar, Persona-AIML [5], desarrolla la personalidad de chat-bots para que cambien el tratamiento a los usuarios de acuerdo a su estado de ánimo y su actitud. Persona-AIML se puso a prueba en una aplicación independiente en Internet Relay Chat y en la web. En tercer lugar, el trabajo en [9] propone un módulo de conversación multimodal basado en la web que detecta cambios de temas solicitados por el usuario y suaviza la transición entre temas solicitando la confirmación del usuario. Nuestra extensión de AIML también incluye nuevas etiquetas para apoyar las conversaciones orientadas a tareas. Sin embargo, en lugar de utilizar un motor de reglas, nuestro bot verifica las respuestas de los usuarios en base a la especificación y el estado actual de la EI. Por otra parte, estas obras tienen interfaces basadas en la Web, en contraste con nuestra interfaz de mundo virtual 3D, que gestiona conversaciones simultáneas con múltiples usuarios.

En la línea de los entornos virtuales en 3D, Komparse [7] implementa un barman como un personaje 3D conversacional en un juego comercial 3D multijugador masivo en línea. El barman recomienda y vende bebidas a los clientes, y también les entretiene con pequeñas conversaciones. La base de conocimientos es una ontología biográfica de celebridades adquirida mediante la aplicación de tecnología de la web semántica. El componente de comprensión del lenguaje natural utiliza una memoria de contexto y un estado del diálogo, y procesa las entradas de los usuarios para seleccionar la respuesta adecuada. Por otra parte, se utiliza un clasificador Bayesiano si la entrada no tiene regla ni patrón asociado. La principal diferencia con nuestra propuesta es que, mientras que Komparse está destinado a entretener a los usuarios en un juego en particular, nuestro sistema modela aplicaciones generales serias, donde los agentes internos apoyan satisfactoriamente la realización, por parte de los usuarios, de tareas estructuradas.

III. ENTORNO 3D, ESTRUCTURADO Y CONVERSACIONAL

Tal y como se ha introducido previamente, el paradigma de las instituciones virtuales permite la creación de Entornos 3D Híbridos y Estructurados. VIXEE [11] es el entorno de

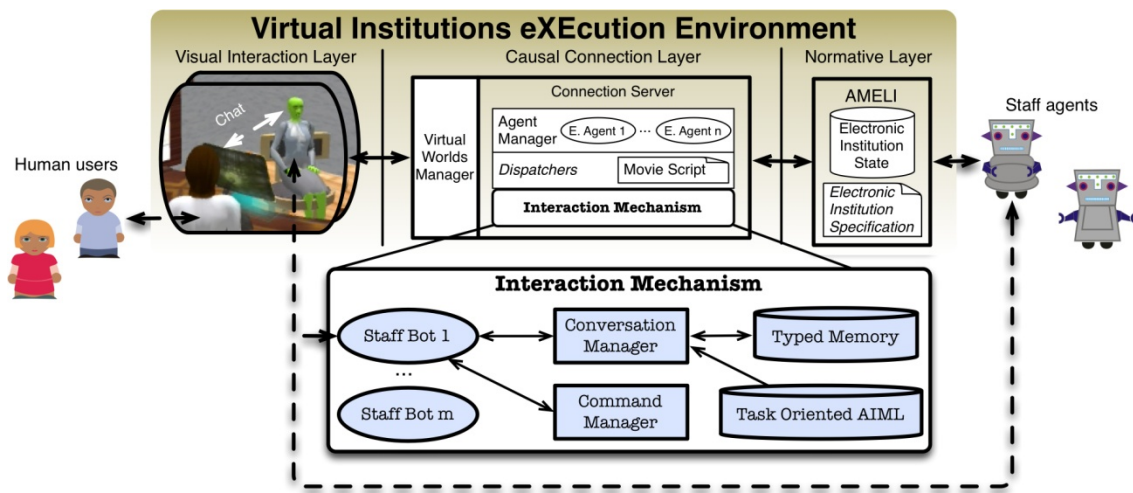


Figura 1. Arquitectura conversacional orientada a tareas.

ejecución de Instituciones virtuales que conecta una Institución Electrónica (EI) con varios mundos virtuales 3D (VW). Éste permite la validación de las interacciones en el VW que tengan significado institucional (es decir, que estén contempladas en la especificación de la EI), y las actualizaciones tanto del VW como del estado de la EI para mantener un estado consistente. Esta sección presenta VIXEE y detalla la incorporación de un nuevo mecanismo de interacción agente-usuario conversacional.

A. Arquitectura

La parte superior de la Fig. 1 representa la arquitectura VIXEE estructurada en tres capas principales: Normativa; de Interacción Visual; y, como middle-ware, la capa de conexión causal.

La **Capa de Interacción Visual** es la interfaz en 3D que representa un espacio inmersivo donde los usuarios pueden interactuar y seguir de forma intuitiva la progresión de las actividades a las que se dedican. Los usuarios humanos (iconos humanos a la izquierda de la Fig. 1) participan en el sistema controlando los avatares (es decir, personajes virtuales 3D) que los representan en el entorno virtual. Además, los agentes internos se visualizan como bots internos en el VW (notar cómo una flecha discontinua en la Fig. 1 enlaza el icono de robot a la derecha con el personaje bot interno).

La **Capa Normativa** a la derecha de la Fig. 1 está compuesta por: AMELI [3]; la especificación de la Institución electrónica (EI); y su estado actual. AMELI es la infraestructura de ejecución de la EI que regula las interacciones de los participantes haciendo cumplir la especificación de la EI en tiempo de ejecución. Brevemente, esta especificación de la EI define: los roles de los participantes institucionales (es decir, internos) y externos; las actividades donde los participantes –jugando roles específicos– realizan tareas; y las regulaciones y protocolos de comunicación asociados a dichas actividades. Los protocolos de comunicación (conversación) se definen como máquinas de

estados finitos, donde los estados representan diferentes etapas de la conversación y las aristas corresponden a ilocuciones (es decir, mensajes institucionales) que los participantes pueden intercambiar. Estas ilocuciones constan de un emisor, un receptor y un contenido, que se expresa en términos de los tipos de la EI (definidos en una ontología) y cuyos valores específicos pueden venir de las entradas de usuario. Por último, los agentes internos (iconos tipo robot en la Fig. 1) son programas de software conectados a AMELI.

La **Capa de Conexión Causal** en la Fig. 1 actúa como middleware. Incluye tanto el gestor del VW (que media toda la comunicación con la plataforma VW) como el servidor extendido de conexión (que lo hace con AMELI). Este último tiene tres componentes: i) el Administrador de agentes, que representa a los humanos como agentes externos (representados como elipses) en AMELI; ii) los despachadores y iii) el mecanismo de interacción extendida. Los despachadores utilizan el llamado mecanismo Movie Script para definir la correspondencia entre eventos AMELI y acciones del VW. Por un lado, un evento generado en AMELI desencadena una acción en el VW, y por lo tanto, la visualización en el VW se actualiza. Por otra parte, para cada acción institucional (regulada por la EI) realizada por un avatar humano en el VW, un despachador genera la ilocución correspondiente en AMELI.

El Mecanismo de Interacción Extendido, que se muestra en la parte inferior de la Fig. 1, soporta las interacciones agente humano. En particular, cada agente interno en AMELI tiene, dentro de este mecanismo de interacción, un bot interno (ver las elipses etiquetadas como *staff bot* en la Fig. 1) que controla el personaje bot interno en el VW y que está dotado de un sistema de comunicación textual. La comunicación agente humano en las versiones anteriores de VIXEE estaba basada en comandos. Este tipo de comunicación se mostró propensa a errores. Por tanto, nuestra propuesta es ampliar el sistema con un sistema conversacional, en el que humano y agente se puedan comunicar mediante lenguaje natural. Ambos sistemas

de comunicación son genéricos para cualquier mundo virtual 3D (soportado por VIXEE) y la especificación de la institución electrónica.

Cuando se usa el sistema basado en comandos, el bot interno entiende los mensajes del usuario como comandos, donde la primera palabra corresponde a la ilocución a ejecutar en el sistema y las palabras sucesivas se mapean al contenido de la ilocución. Concretamente, el bot interno filtra los mensajes de los usuarios mediante la comparación de la primera palabra con las ilocuciones permitidas en la institución electrónica (EI) y utiliza un Administrador de Comandos para validar su contenido estructurado. Los comandos son entonces directamente mapeados a ilocuciones dirigidas a agentes internos en AMELI. En la dirección inversa, una ilocución que el agente interno envía al usuario se traduce a un mensaje enviado por el bot personal al usuario en el entorno virtual.

Nuestro interés se centra en el sistema de conversación, que presenta la complejidad intrínseca de los diálogos en lenguaje natural, donde los usuarios pueden expresar sus intenciones de diferentes formas, y los bots internos tienen que entender aquello que dicen los usuarios y así darles soporte en su tarea. Para ello, un bot interno debe: i) identificar la tarea del usuario; ii) solicitar al usuario, en caso de que sea requerida, toda aquella información que sea necesaria para completar la tarea; iii) validar las entradas de usuario con respecto a la especificación y al estado actual de la EI y guardarlas en una memoria que almacene diferentes tipos de datos; y iv) enviar una ilocución al agente interno para que la tarea del usuario se pueda completar. En la dirección inversa, cada ilocución del agente interno enviada al usuario debe ser expresada en lenguaje natural.

Este proceso bidireccional requiere de un agente interno para controlar dinámicamente el flujo y el estado de una conversación orientada a tareas con múltiples usuarios. Dicha conversación está basada en i) las entradas del usuario, ii) las ilocuciones del agente interno, y iii) la especificación y estado actual del sistema. Los agentes internos delegan en un Gestor de la Conversación: i) la actualización de los estados de las conversaciones, ii) el almacenamiento de las respuestas de los usuarios en una memoria que considera la ontología definida en la especificación de la institución electrónica, y iii) la interpretación del conocimiento conversacional, que está basado en AIML (Lenguaje de Inteligencia Artificial de Marcado). Tal y como se ha mencionado, AIML ha sido concebido para programar robots conversacionales reactivos que siguen conversaciones sencillas con los usuarios. Nosotros extendemos esta funcionalidad básica de AIML y proponemos el denominado AIML orientado a tareas, que incluye etiquetas especiales para permitir conversaciones estructuradas dentro de nuestro sistema. La siguiente sección explica con más detalle esta extensión y las interacciones entre los diferentes participantes de las conversaciones orientadas a tareas.

IV. CONVERSACIONES ORIENTADAS A TAREAS

AIML se concibió con el objetivo de crear bots conversacionales reactivos que puedan entablar diálogos

sencillos en lenguaje natural cuando así les sea solicitado por los usuarios. De hecho, AIML es un dialecto XML que encapsula el conocimiento conversacional en objetos de datos llamados categorías. Básicamente, cada categoría se define por una cuestión de entrada o pregunta del usuario (es decir, un patrón o *pattern* para la entrada de usuario) y una respuesta de salida (una plantilla o *template* para generar la respuesta del bot). Las categorías se agrupan en tópicos (*topic*) y pueden especificar un cambio de tema. Por lo tanto, los diseñadores de bots pueden encapsular el conocimiento conversacional en tópicos y fijar un flujo de conversación. A pesar de su sencilla estructura y funcionamiento, los robots conversacionales AIML pueden dar una respuesta a casi cualquier frase que el usuario pueda imaginar, es una cuestión de ampliar el conocimiento de la conversación con más y más ficheros AIML.

Sin embargo, AIML no funciona bien en un entorno 3D estructurado donde los bot internos, más allá de tener habilidades conversacionales básicas para dar la bienvenida y despedida al usuario, necesitan ser proactivos con el fin de gestionar conversaciones que apoyen al usuario en la realización de sus tareas. La sección anterior introdujo el Mecanismo de Interacción, que se ha ampliado para permitir conversaciones humano-bot en lenguaje natural. Esta sección proporciona más detalles sobre la estructura y el despliegue de estas conversaciones y propone *AIML orientado a tareas*, una extensión de AIML para apoyar las conversaciones orientadas a tareas.

A. Estructura de la conversación

La estructura de una conversación orientada a tareas tiene tres etapas: bienvenida, realización de una tarea y despedida. Tanto la bienvenida como la despedida se articulan mediante la definición de un tema/tópico de *welcoming* y se define, además, un tópico para cada tarea soportada. El *welcoming* es el tópico por defecto del bot interno, usa etiquetas estándar de AIML, y se utiliza para saludar y despedir a los usuarios e informarles sobre la actividad. Por otro lado, el flujo de la conversación que corresponde a un tópico de tarea (*task*) se inicia cuando en la etapa de bienvenida el bot reconoce que el usuario está solicitando la realización de la tarea correspondiente. Dicho flujo sigue la estructura definida por la máquina de estados finita que se muestra en la Fig. 2 y se articula en base a dos tipos de interacciones usuario-agente que se realizan de forma sucesiva: i) la primera interacción es iniciada por el usuario para la realización de una tarea; y ii) la segunda interacción es iniciada por el agente, y contiene la confirmación de que dicha tarea se haya completado o no con éxito.

El primer tipo de interacción es más complejo, y comienza tras el reconocimiento por parte del bot interno de que una entrada de usuario en la etapa de bienvenida corresponde a una petición de realización de una tarea. Esto se hace mediante el uso de categorías estándar de AIML, que redirigen entradas concretas (que contienen determinadas palabras clave) de usuarios a un tópico de tarea. En ese punto, el bot interno toma la iniciativa y pide al usuario los datos necesarios

para completar la tarea (ver estado *A asked* en la Fig. 2) y espera la respuesta del usuario para ser procesada en el estado *R responded*. Si la entrada de usuario no es válida, el bot interno informa al usuario, y el estado de la conversación pasa a *F failed*. En este estado, el bot interno repite la cuestión (alcanzando de nuevo el estado *A asked*) hasta que el usuario solicita la cancelación (estado *CR cancel requested*), provocando que el bot interno finalice la tarea. Por otro lado, si toda la información se recoge de forma correcta, el bot interno termina las preguntas (estado *T terminated*) presentando al usuario la información que ha obtenido y requiriéndole su confirmación antes de ejecutar la ilocución en el sistema. En este punto, el usuario puede requerir la cancelación con las mismas consecuencias que para el estado mencionado previamente *F failed*. En caso contrario, el bot interno ejecuta la ilocución dentro del sistema, informa de ello al usuario y cambia el tópico a *wellcoming*.

El bot interno es por lo tanto un autómat que selecciona la siguiente acción a ejecutar basándose en las entradas del usuario, en las ilocuciones que se definen en la especificación de la Institución Electrónica (EI), y en los valores permitidos del contenido de dichas ilocuciones en el estado actual del sistema. La Fig. 3 representa el Diagrama de secuencia de VIXEE correspondiente a la interacción humano-agente. Cada vez que un usuario envía un mensaje al bot interno, éste último obtiene del gestor conversacional: la tarea actual del usuario, el tipo esperado en la respuesta del usuario (si lo hay), y el estado actual de la conversación. A continuación, procesa la entrada del usuario y calcula el siguiente patrón AIML. En primer lugar, si el estado es *R responded*, el bot interno intenta reconocer un valor en la entrada (o mensaje) del usuario. Para ello, obtiene la definición del tipo relacionado en la especificación de la EI y los valores permitidos en tiempo de ejecución desde el estado de la EI, e intenta encontrar ese valor en la entrada del usuario. Si lo encuentra, envía el valor válido al gestor conversacional, que almacena dicho valor en su memoria. Después, si el estado de la conversación es *C confirmed*, el bot ejecuta la correspondiente ilocución donde el remitente es el usuario, el receptor es el agente interno y el contenido son los datos almacenados que el bot ha preguntado al usuario.

Finalmente, el bot interno, basándose en el resultado del proceso anterior y en el estado actual de la conversación, computa el patrón que conducirá al siguiente estado y lo envía al gestor conversacional. El gestor conversacional interpreta el AIML orientado a tareas con el patrón (mensaje del usuario) dado, actualiza el estado de la conversación, y devuelve el mensaje de respuesta (frase) al bot, que es el quién lo enviará al usuario. Cabe notar, no obstante, que en caso de que la ilocución haya sido emitida con éxito, el mensaje de respuesta informará de ello, pero no del éxito de completado de la tarea concreta.

Llegado este punto, el sistema queda a la espera de que AMELIE ejecute la ilocución y sea el agente interno quien informe del resultado al usuario. Esto dará pie al inicio del segundo tipo de interacción: la iniciada por el agente. La conversación en este caso es más simple, ya que el bot interno

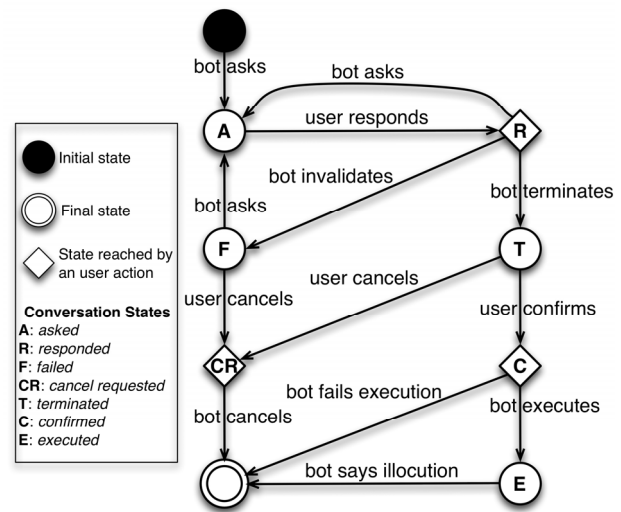


Figura 2. Máquina de estados finita que describe los estados de la conversación para un tópico de tarea.

se limita a procesar la ilocución que el agente interno ha enviado al usuario dentro de AMELI y envía el resultado al usuario en lenguaje natural mediante una ventana de chat en el la interfaz 3D. Concretamente, la Fig. 4 muestra cómo el bot interno recibe la ilocución del agente interno de AMELI y computa el patrón (*pattern*) que envía al gestor de la conversación. Después, el gestor conversacional accede al AIML orientado a tareas para obtener la plantilla (*template*) correspondiente, actualiza el estado de conversación y envía de vuelta el mensaje resultante al bot interno. Por último, el bot interno envía al usuario un mensaje de chat que representa la ilocución en lenguaje natural y la conversación de la tarea alcanza su estado final.

La siguiente subsección detalla cómo hemos extendido AIML para poder llevar a cabo la estructuración de las conversaciones.

B. Conocimiento AIML orientado a tareas

Como se mencionó previamente, el conocimiento que representa una tarea se encapsula en un único tópico. Además, hemos extendido AIML de forma que el gestor conversacional es capaz de controlar el flujo de la conversación. Concretamente, hemos incluido dos nuevas etiquetas AIML, que están localizadas dentro de la etiqueta *template: taskresptype* y *taskstate*. A continuación mostramos un ejemplo general:

```

...
<template>
<taskstate>state</taskstate>
<taskresptype>responsetype</taskresptype>
...
expression
</template>
...

```

Primero, el formato (o tipo) esperado de la respuesta del usuario se indica con la etiqueta *taskresptype*. El valor, *responsetype*, corresponde con un tipo especificado en la ontología de la EI.

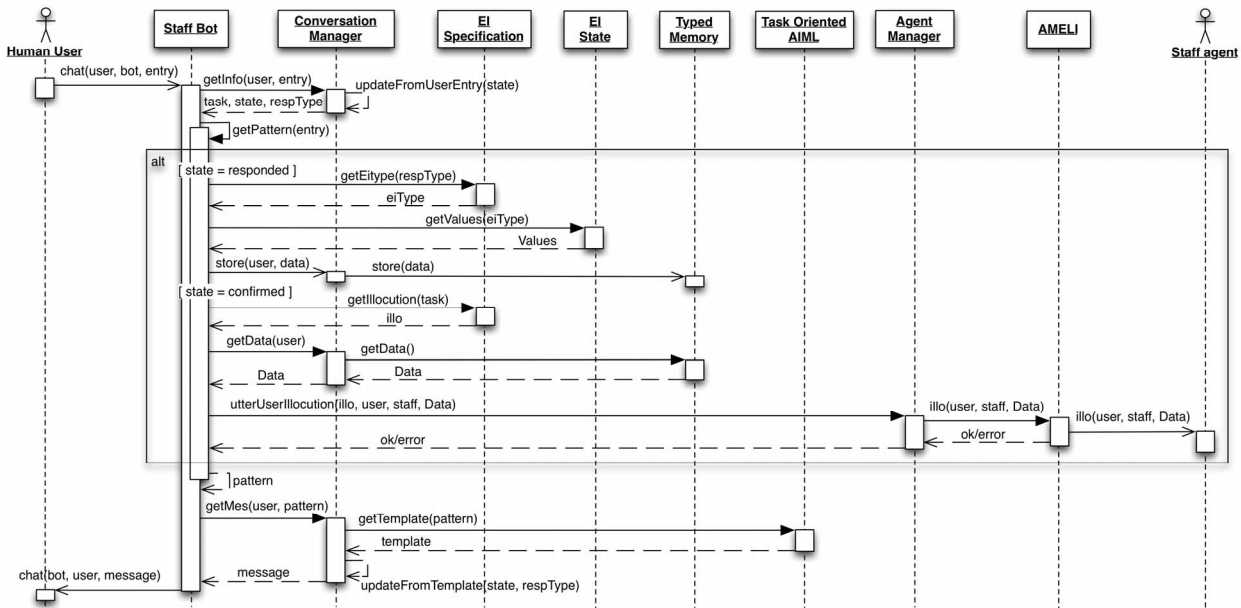


Figura 3. Diagrama de secuencia que describe la interacción de usuario a agente (iniciada por el usuario).

Segundo, *taskstate* indica el estado de la conversación orientada a tareas, y puede ser uno de los estados de conversación definidos en la Fig. 2. El agente interno es capaz de redireccionar la conversación al estado deseado porque las categorías que representan los estados de la conversación tienen unos patrones definidos. En particular, el estado final cambia el tópico de la conversación a *welcoming*. Esto se ha realizado con AIML estándar, añadiendo la etiqueta *think* dentro de la etiqueta *template* como en el siguiente ejemplo:

```

...
<think><set
name="topic">welcoming</set></think>
...

```

Siguiendo este esquema, la Fig. 5 muestra parte de una conversación real mantenida en el prototipo v-mWater (que se introduce en la siguiente sección), donde un usuario conversa con el Bot de registro para llevar a cabo una tarea de registro. Concretamente, el usuario registra un derecho de agua con identificador *wr1* a un precio de 25 euros. También muestra un extracto del código AIML que permite tal conversación. Las categorías en el tópico *welcoming* cambian al tópico de tarea *register* cuando el usuario menciona “register”, “registration” o cualquier palabra derivada de las mismas. Además, este fragmento considera el estado de la tarea *asked* y los tipos *wrid* (identificador de derecho del agua) y *price* (precio) de la ontología de la institución electrónica. Éstos son los tipos de datos que corresponden a los datos que el usuario proporciona al bot interno durante la conversación.

V. EVALUACIÓN

En esta sección se evalúa la usabilidad del mecanismo conversacional propuesto comparándola con la de la aproximación basada en comandos. En concreto, hemos

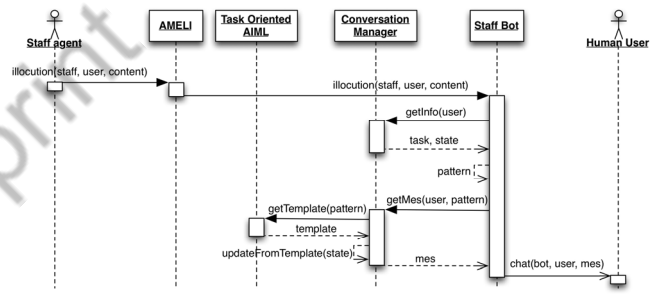


Figura 4. Diagrama de secuencia que describe la interacción de agente a usuario (iniciada por el agente).

realizado tests de usuario utilizando el prototipo v-Water, un entorno virtual 3D estructurado que implementa un mercado virtual del agua para negociar derechos de agua. A continuación se definen los objetivos del test, posteriormente se describe la metodología de test, y finalmente se presentan y discuten los resultados.

A. Objetivos del test

El objetivo principal del test es evaluar la usabilidad general de la propuesta de conversación orientada a tareas en entornos 3D híbridos y estructurados. Para ello, nos centramos en diferentes criterios de usabilidad tales como la eficacia, eficiencia, los errores y la satisfacción, y los comparamos con el enfoque basado en comandos. También aspiramos a abrir un debate sobre cómo afecta la experiencia de los usuarios con los ordenadores en su experiencia con los sistemas basados en comandos y los sistemas conversacionales. Además, este estudio de usabilidad nos permitirá detectar problemas de diseño -tanto en la estructura como en el contenido- de conversaciones orientadas a tareas en entornos virtuales 3D estructurados.



Figura 5. Extracto de una conversación orientada a tarea para el registro de un derecho de agua, y su código AIML relacionado.

TABLA I. DISEÑO DEL EXPERIMENTO ENTRE SUJETOS.

| Participantes | Tarea 1, Tarea 2 | Tarea 1, Tarea 2 |
|---------------|------------------|------------------|
| P1-P5 | Conversacional | Comandos |
| P6-P10 | Comandos | Conversacional |

TABLA II. PREGUNTAS DEL CUESTIONARIO POST TEST.

| | |
|--|---|
| Q1 | No sentí que necesitara ayuda al hablar con el bot. |
| Q2 | No me sentí frustrado/a mientras hablaba con el bot |
| Q3 | Lo que el bot me dijo tenía sentido |
| Q4 | Sabía qué responder al bot |
| Q5 | ¿Cómo de cómoda fue la comunicación? |
| Q6 | Sentí que el bot me entendió |
| Q7 | El robot daba las respuestas que me esperaba |
| Las posibles respuestas son: 1: Nunca / muy incómodo; 2: A veces / Incómodo - 3: Regularmente / normal; 4: A menudo / Cómodo - 5: Siempre / Muy cómodo | |

B. Metodología

En la realización de los tests hemos seguido los métodos de evaluación sumativo y comparativo [2]. El método sumativo se centra principalmente en recoger datos cuantitativos relacionados con la usabilidad de la aproximación conversacional. Para la evaluación comparativa hemos realizado un diseño de test denominado *within-subject*, donde cada usuario prueba cada una de las aproximaciones (conversacional y basada en comandos) y medimos el rendimiento del usuario en cada aproximación.

Se reclutaron 10 participantes, la mitad de ellos fueron seleccionados con experiencia previa con sistemas de interacción basados en comandos, y la otra mitad eran usuarios noveles. Los participantes formaban una población diversa en términos características tales como la edad, el sexo, y la ocupación.

Tal como indica la Tabla I, se pidió a todos los participantes repetir las mismas dos tareas utilizando tanto interacciones conversacionales como basadas en comandos. Para mitigar el efecto “arrastre”, la mitad de los participantes

comenzaron utilizando la interacción basada en comandos, mientras que la otra mitad se inició con la conversacional.

Concretamente, se pidió a los usuarios realizar las siguientes tareas, literalmente descritas como:

- Tarea 1: “Tu objetivo es preguntar al Information Manager acerca de las últimas 2 transacciones en el mercado”.
- Tarea 2: “Tu objetivo es solicitar al Registration Manager el registro de un derecho de agua, identificado como wr1, por un precio de 25 euros”.

Cabe notar que los usuarios realizaron las tareas en este mismo orden: la tarea 1 seguida de la tarea 2. La razón era que la primera tarea es una poco más simple que la segunda y, por lo tanto, se asumió que los usuarios noveles (principiantes) en interacciones basadas en comando encontrarían menos dificultades (es decir, tendrían menos frustraciones) teniendo primero que realizar esta tarea.

El equipo de evaluación estuvo compuesto por un moderador y un observador. El moderador guió al usuario (en caso de necesidad), le introdujo el test y le dio el formulario de consentimiento, la descripción de las tareas, y el cuestionario posterior a la prueba. El observador tomó notas durante la prueba. Las pruebas se llevaron a cabo en la casa o en la oficina de los usuarios. El material que se usó fue un ordenador que ejecutaba tanto el servidor del mundo virtual como la aplicación cliente del mundo virtual. También se registraron las interacciones del usuario y el sonido.

El protocolo del test consta de 4 fases. En la primera fase, la entrevista *pre-test*, donde dimos la bienvenida al usuario, explicamos los objetivos de la prueba, y le preguntamos acerca de su experiencia con sistemas de interacción basados en comandos o conversacionales. En la segunda fase, el *entrenamiento*, el usuario practicó a través de una demostración para aprender a moverse en el entorno 3D y interactuar en el mismo. Esta parte de entrenamiento fue totalmente guiada, excepto al final, cuando el usuario pudo moverse e interactuar libremente en el escenario de la demostración. La tercera fase fue el *test*, el usuario realizó las tareas del test sin recibir orientación a menos que se quedara sin recursos. Finalmente, el usuario contestó un cuestionario

post-test con preguntas tanto cualitativas como cuantitativas, incluyendo una última pregunta abierta para cualquier comentario adicional que quisiera realizar.

C. Resultados y discusión

En esta sección se analizan los resultados del test y se discute sobre si se han conseguido los objetivos del test que, tal y como se dijo anteriormente, se centra principalmente en comparar diferentes criterios de usabilidad en las aproximaciones de interacción basadas en comandos y conversacional, así como en cuantificar la influencia del perfil de usuario en el logro de tareas. Los datos provienen de: un cuestionario post-test rellenado por el usuario una vez acabado el test, comentarios de usuarios, notas del observador, y la revisión de las grabaciones de voz y de escritorio que se realizaron mientras que los participantes realizaban la tarea.

La Tabla II resume las siete preguntas incluidas en el cuestionario de satisfacción realizado al final del test, y la Fig. 6 recoge las respuestas de los usuarios. En la figura se puede ver que el eje X muestra cada una de las preguntas y el eje Y muestra los valores medios de las respuestas, en una escala Likert de cinco puntos. Las preguntas se formulan de manera

Cinco preguntas del post-cuestionario tuvieron respuesta doble, una para la aproximación conversacional y otra para la basada en comandos. Sólo dos de las preguntas requirieron de una única respuesta relacionada con la aproximación conversacional. Por lo tanto, la gráfica de barras de la Fig. 6 muestra cinco pares de barras, las de color azul marino corresponden a la conversacional y las azul claro representan los datos del sistema basado en comandos. En general, los resultados cuantitativos que se obtuvieron de estas cinco primeras preguntas fueron satisfactorios, y la respuesta media del sistema conversacional fue superior al basado en comandos. Las preguntas individuales (Q6, Q7) son preguntas relacionadas con la capacidad del bot conversacional tanto para entender al usuario como para proporcionar respuestas entendibles por el usuario. Éstas presentan promedios de 4,7 y 4,6 respectivamente, por lo que podemos afirmar que obtienen también buenos resultados.

En cuanto a la eficacia de las tareas, la interacción conversacional obtuvo una tasa de finalización de la tarea del 100%. Ese no fue el caso para la aproximación basada en comandos, donde el 30% de los participantes fracasaron en la tarea 1 y el 20% lo hicieron en la tarea 2. Consideramos un fracaso aquella tarea que el participante no pudo completar sin la ayuda del moderador o cuando la tarea se realizó sin éxito (por ejemplo, cuando se registra a un precio incorrecto). Adicionalmente, los usuarios realizaron las tareas con un promedio de error menor (0,3 errores en promedio) en el sistema conversacional que en el sistema basado en comandos (2,2 errores en promedio). Dicha diferencia se ha mostrado significativa en un t-test con un p-value de 0,01.

Respecto a la eficiencia, mostramos los resultados basándonos en el número de mensajes que necesitaron enviar los usuarios a los bots conversacionales para completar las tareas asignadas, ya sea mediante conversación en lenguaje natural o mediante comandos. Los usuarios enviaron un promedio de 6,7 mensajes utilizando los comandos y 9,7 mensajes al conversar utilizando lenguaje natural. Si

que 1 corresponde a la respuesta más negativa y 5 a la más positiva.

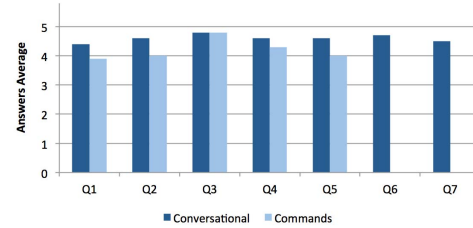


Figura 6. Resultados del cuestionario post-test. Eje X: preguntas de la Tabla II, Eje Y: valores medios de las respuestas.

analizamos estos promedios respecto al número mínimo de mensajes necesarios para ambas tareas en comandos (este mínimo era de 4 mensajes), y el mínimo de mensajes para conversacional (que era 9), éstos representan un 168% en el sistema de comandos, y sólo un 107% en el sistema conversacional. Estos resultados muestran que los usuarios que interactuaron utilizando lenguaje natural dedicaron menos esfuerzo al perseguir su objetivo. Esto también se ve corroborado por el menor número de errores cometidos al conversar con los bots.

Si analizamos los resultados en base a la experiencia del usuario en sistemas de comandos (es decir, experto o principiante), el post-cuestionario de satisfacción muestra que los expertos califican de manera casi idéntica ambos métodos (conversación con un promedio de 4,35 y comandos con 4,57) mientras que para los principiantes el lenguaje natural obtiene un promedio de 4,65 en comparación con el promedio de 3,63 obtenido para el sistema de comandos, con p-valor = 0,014. Este resultado denota que los expertos se sienten cómodos con ambos sistemas de interacción, pero sin embargo ése no es el caso para los principiantes. Cuando se trata del analizar el número de errores en la interacción basada en comandos, la diferencia entre los noveles (con una media de 3,8) y expertos (con una media de 0,6) demostró ser significativa con un p-valor de 0,01. Además, los principiantes envían más mensajes que los expertos, con un promedio de 8,4 y 5 respectivamente, de nuevo con un p-valor de 0,01. Si analizamos los resultados obtenidos en la aproximación conversacional, los datos recogidos muestran que tanto los expertos como los noveles cometieron un número similar de errores y enviaron un número similar de mensajes, lo que demuestra que ambos perfiles de usuario se comportan de forma similar en la aproximación conversacional. Este análisis de datos sugiere el uso de una interacción multimodal, facilitando la coexistencia de los dos estilos de interacción.

Los test de usuario también tenían como objetivo detectar fallos, deficiencias o inconsistencias en la definición del AIML orientado a tareas, ya que pueden afectar negativamente a la interacción usuario-bot. Tal y como se mencionó previamente, las conversaciones orientadas a tareas se estructuran en etapas de bienvenida, de tarea y de despedida. Sin embargo, el bot interno no dio la bienvenida al usuario de forma proactiva, es decir, el bot esperó a que el usuario tomara la iniciativa en el saludo. Por lo tanto, algunos usuarios que se apresuraron a preguntar al bot acerca de la tarea que querían realizar, recibieron como respuesta un saludo, haciendo necesario para los usuarios repetir su petición.

Por otra parte, algunos usuarios se confundieron debido a una expresión interrogativa utilizada por el bot en relación a las últimas transacciones, ya que los usuarios suponían que tenían que proporcionar identificadores de transacción, que no eran conocidos por ellos, en lugar del número de transacciones (es decir, cuántas). Por tanto, el conocimiento (AIML) de los bots se debería revisar para evitar así estas confusiones u otras similares. Por último, cuando una tarea requería del usuario introducir varios datos, algunos usuarios intentaban proporcionar todos los datos en una única frase. Pensaban que el bot podría entender toda la frase, pero éste no fue el caso, por lo que éste constituye otro punto a tener en cuenta en la revisión del AIML del bot.

VI. CONCLUSIONES

Las Instituciones Virtuales son entornos 3D estructurados e híbridos donde los participantes (tanto humanos como agentes software) realizan actividades serias. Estas actividades se especifican (por ejemplo, en términos de los roles implicados o de los diálogos de los participantes) en tiempo de diseño y se aplican en tiempo de ejecución. Los agentes internos son agentes institucionales encargados de hacer posible las actividades de la organización. Por tanto, en muchas ocasiones los usuarios deben interactuar con los bots internos (la representación 3D los agentes internos) que los apoyan en la consecución de su tarea.

En este trabajo se ha presentado la integración de un nuevo mecanismo de conversación para facilitar la interacción usuario-agente en entornos 3D estructurados. Para ello, hemos propuesto una extensión de AIML para tratar conversaciones orientadas a la tarea, que se basan en la especificación de las tareas y el estado actual del sistema. Hemos evaluado el nuevo mecanismo de conversación en v-mWater, una institución virtual para el comercio de derechos de agua. Los resultados han mostrado buenas medidas de usabilidad en términos de eficiencia, eficacia y satisfacción del usuario en el enfoque conversacional. También hemos comparado con otro estilo de interacción, ya incorporado en la infraestructura, basado en comandos. En el cuestionario post-test de satisfacción el enfoque conversacional ha sido mejor valorado que el basado en comandos. Sin embargo, un análisis más detallado de los datos, basado en las habilidades de los usuarios (tanto en el sistema conversacional como en el basado en comandos), nos sugiere la coexistencia de ambos estilos de interacción. Como trabajo en curso, estamos colaborando con lingüistas para rediseñar el conocimiento AIML con el fin de mejorar los resultados de pruebas futuras. Asimismo, planeamos incorporar conversaciones de voz sintetizada que faciliten aún más las tareas de los usuarios.

AGRADECIMIENTOS

Trabajo financiado por CSD2007-0022, TIN2011-24220 y TIN2012-38876-C02-02).

REFERENCIAS

- [1] P. Almajano, E. Mayas, I. Rodríguez, M. Lopez-Sanchez, and A. Puig. Structuring interactions in a hybrid virtual environment: Infrastructure & usability. In GRAPP'13, pages 288-297, 2013.
- [2] D. Bowman, J. Gabbard, and D. Hix. A survey of usability evaluation in virtual environments: classification and comparison of methods. Presence: Teleoperators & Virtual Environments, 11(4):404-424, 2002..

- [3] M. Esteva, B. Rosell, J. A. Rodríguez-Aguilar, and J. L. Arcos. AMELI: An agent-based middleware for electronic institutions. In AAMAS'04, pages 236-243, 2004.
- [4] J. Ferber, O. Gutknecht, and F. Michel. From agents to organizations: An organizational view of multi-agent systems. In Agent-Oriented Software Engineering IV, volume 2935 of Lecture Notes in Computer Science, pages 214-230. Springer Berlin Heidelberg, 2004.
- [5] A. M. Galvao, F. A. Barros, A. M. Neves, and G. L. Ramalho. Persona-aiml: An architecture developing chatterbots with personality. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3, pages 1266-1267. IEEE Computer Society, 2004.
- [6] A. C. Graesser, P. Chipman, B. C. Haynes, and A. Olney. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. Education, IEEE Transactions on, 48(4):612-618, 2005.
- [7] T. Kluwer, F. Xu, P. Adolphs, and H. Uszkoreit. Evaluation of the kompare conversational non-player characters in a commercial virtual world. In LREC, pages 3535-3542, 2012.
- [8] F. A. Mikic, J. C. Burguillo, M. Llamas, D. A. Rodríguez, and E. Rodríguez. Charlie: An aiml-based chatterbot which works as an interface among innes and humans. In EAEEIE Annual Conference, 2009, pages 1-6. IEEE, 2009.
- [9] K. Mori, A. Jatowt, and M. Ishizuka. Enhancing conversational flexibility in multimodal interactions with embodied lifelike agent. In Proceedings of the 8th international conference on Intelligent user interfaces, pages 270-272. ACM, 2003.
- [10] J. R. Searle. Speech acts: An essay in the philosophy of language, volume 626. Cambridge university press, 1969.
- [11] T. Trescak, I. Rodríguez, M. Lopez Sanchez, and P. Almajano. Execution infrastructure for normative virtual environments. Engineering applications of artificial intelligence, 26(1):51-62, 2013.
- [12] R. Wallace. The anatomy of a.l.i.c.e. In R. Epstein, G. Roberts, and G. Beber, editors, Parsing the Turing Test, pages 181-210. Springer Netherlands, 2009.
- [13] F.-Y. Wang, P. B. Mirchandani, and Z. Wang. The vista project and its applications. Intelligent Systems, IEEE, 17(6):72-75, 2002.



Pablo Almajano estudió Ingeniería Informática en la Universidad de Comillas (España), realizó el máster en visión por computador en la Universidad Autónoma de Barcelona y se doctoró en el Instituto de Investigación en Inteligencia Artificial (IIIA-CSIC) en el año 2014. Su principal interés de investigación se centra en el soporte a la decisión en sistemas multi-agente híbridos y estructurados.



Maite López-Sánchez es profesora titular en la Universidad de Barcelona (España) y coordinadora en la UB del máster interuniversitario en Inteligencia Artificial. Anteriormente, fue responsable de investigación en la empresa iSOCO e investigadora visitante en la University of Southern California (USA). El doctorado lo realizó en el Instituto de Investigación en Inteligencia Artificial (IIIA-CSIC). A lo largo de su investigación, ha trabajado en diferentes áreas de la IA, tales como organizaciones de sistemas multi-agente, aprendizaje, robótica, negociación o comercio electrónico.



Inmaculada Rodríguez se doctoró por la Universidad de Alcalá (España) en 2004 y se licenció en Informática por la universidad de Granada. Actualmente es profesora titular en la Universidad de Barcelona y miembro del grupo de investigación WAI (Volume Visualization and Artificial Intelligence). Su investigación se centra en animación por computador, integración de IA en mundos virtuales 3D, serious games y gamification.



Enric Mayas se graduó en Ingeniería Informática por la Universidad de Barcelona (España) en el año 2013, fue estudiante erasmus de Chalmers University of Technology (SE) y actualmente es estudiante de máster en La Salle. La interacción persona-máquina constituye la principal área de investigación que es de su interés.