Incorporación de interacción vocal en mundos virtuales usando VoiceXML

César González Ferreras, Arturo González Escribano,
David Escudero Mancebo y Valentín Cardeñoso Payo
Departamento de Informática
Universidad de Valladolid
e-mail: descuder@infor.uva.es

Resumen

En este trabajo se presenta un marco de desarrollo de aplicaciones diseñado para facilitar la incorporación sistemática de interacción vocal a mundos virtuales. Empleando una plataforma de gestión de diálogo basada en VoiceXML, se propone una arquitectura de referencia que permite integrar fácilmente guiones de diálogo en aplicaciones de interacción con mundos virtuales, explotando las posibilidades que proporciona el estandar EAI para VRML. Se contempla la combinación de los diálogos hablados, especificados en VoiceXML, con la interacción habitualmente soportada por las consolas de visualización de VRML con el fin de soportar la interacción multimodal adecuada para la construcción de entornos virtuales de naturalidad adecuada.

Palabras clave: interacción multimodal, realidad virtual, sistemas de diálogo, avatares.

1. Introducción

La realidad virtual (RV) ha demostrado ser un instrumento de gran utilidad en determinadas aplicaciones de simulación, educación y entretenimiento. Un factor determinante del éxito de las aplicaciones de RV es el grado de naturalidad de la interacción que sean capaces de proporcionar al usuario, pues de ella depende en gran medida el nivel de inmersión que se proporciona al usuario. Una interacción que quiera ser natural, necesita emplear necesariamente el canal sonoro de forma coordinada con el canal visual. El habla es el instrumento básico y genuino de comunicación del ser humano con su entorno empleando un canal sonoro, por lo que es lógico pensar

*Este trabajo ha sido financiado parcialmente por la Junta de Castilla y León, en el marco del proyecto VA083/03 de la Consejería de Educación y por el MCyT en el marco del proyecto TIC2003-08382-C05-03.

que su incorporación en aplicaciones de RV pueda aportar beneficios esenciales a la naturalidad de la interacción.

En la bibliografía pueden encontrarse diversas propuestas de incorporación de habla en aplicaciones de RV o de animación 3D, en un intento de aumentar las capacidades de interacción persona-computador. Existen dos familias básicas de aplicaciones de este tipo: las aplicaciones de Habla Visual (o VS, del inglés *visual speech*) y las de control o navegación vocal de mundos virtuales. En las aplicaciones de VS se incorpora un personaje o un rostro que, con mayor o menor grado de realismo, articula representaciones visuales de los sonidos significativos elementales (visemas) sincronizadas con la componente sonora del mensaje (ver [1]). El objetivo fundamental de este tipo de aplicaciones es mejorar el grado de inteligibilidad del mensaje, por un lado, y proporcionar un mecanismo extra de inserción de información emocional gráfica, por otro. En ellos, el guión de acciones o movimientos del personaje puede venir anotado sobre la representación simbólica o sonora del mensaje hablado, empleando habitualmente soluciones desarrolladas 'ad hoc' con una metodología prueba-error. En las aplicaciones de control o navegación vocal de mundos virtuales, se proporciona al usuario un lenguaje artificial de gobierno de la configuración del mundo que le permite, empleando el habla, alterar el estado del mismo (como por ejemplo en [2]).

Sin embargo, el nivel de integración de las posibilidades que brindan en la actualidad las Tecnologías del Habla para el diseño de sistemas de diálogo dista mucho de ser el más adecuado en este tipo de aplicaciones, a nuestro juicio. Por un lado, ambas familias de aplicaciones rara vez se armonizan sobre una plataforma vocal única en la que se gestione adecuadamente la conversión texto-voz y el reconocimiento de habla. Por otra parte, parecen perseguir objetivos complementarios en tanto que se especializan bien en proporcionar una experiencia mejorada de presentación de información o un canal de entrada más natural, dejando de lado la inevitable coordinación entre ambas fases de interacción. Este tipo de enfoques son los que se detallan en trabajos como los de Traum [4] o McGlashan [5].

En este trabajo, planteamos la incorporación del canal hablado a la interacción con el VR de una forma integral, basándonos en un estándar de especificación de diálogos que permite construir de manera sencilla el guión de los procesos de interacción entre el usuario y los personajes u objetos inmersos en un mundo virtual. La especificación de la interacción por medio de diálogos hablados permite combinar adecuadamente los beneficios de las aplicaciones de VS y de control vocal, si se dispone de una plataforma de implementación vocal que explote adecuadamente el canal visual de presentación y el canal sonoro de entrada dentro de un navegador de VR. Esta aproximación al problema supone, por otra parte, un paso adelante para la planificación de objetivos y acciones asociadas a la interacción, en un nivel conceptual adecuado alejado de los detalles de implementación.

VoiceXML [3] es un estándar que define una aplicación XML para la especificación de diálogos hablados usuario-sistema. Aunque fué introducido originalmente como lenguaje de marcas para la construcción de sistemas de respuesta telefónica avanzada,

sus capacidades exceden claramente este marco y es posible emplear intérpretes de VoiceXML que se enlacen con plataformas de implementación vocal en entornos de sobremesa o, como comentamos en este trabajo, plataformas específicas de comunicación con consolas de navegación VR. Aunque la estructura básica del lenguaje está diseñada pensando en un modelo de diálogo con iniciativa dirigida por el sistema que se pueda representar formalmente por medio de máquinas con un número finito de estados, las posibilidades de programación de scripts tanto para el despacho de eventos como para el procesamiento avanzado de las variables de control de estado o para incorporar interacción con otros canales alternativos al sonoro abren un enorme abanico de técnicas para el soporte de interacción multimodal en VoiceXML, si bien no todas con el mismo nivel de estructuración de la solución.

En esta comunicación definimos un marco de trabajo para la especificación de escenarios de interacción en los que las componentes visuales (estáticas o dinámicas) se representan en VRML y la interacción entre el usuario y los personajes u objetos presentes en el mundo se especifican por medio de guiones escritos en VoiceXML. Estos guiones –o diálogos– permiten describir la secuencia de turnos de usuario y de sistema y representar de forma homogénea el conjunto de acciones que gestionarán los cambios en el mundo 3D desencadenados por dichos turnos. La comunicación entre la descripción VRML y la especificación VoiceXML se realiza siguiendo el estándar EAI.

El resto de la comunicación está estructurado como sigue. En primer lugar discutimos las ventajas que puede aportar la multimodalidad basada en diálogos a la interacción con mundos 3D. Luego, se describe el marco de trabajo propuesto y se proporciona una guía para su uso en el desarrollo de aplicaciones de RV que aúnen VRML y VoiceXML. Esta guía de desarrollo se ilustra con una aplicación sencilla. Finalmente, se presentan las conclusiones fundamentales de nuestro estudio y se apuntan algunas líneas de trabajo futuro.

2. Sistemas de Diálogo y Mundos 3D

La interacción multimodal supone poner en juego al menos dos modos de interacción entre el hombre y la máquina: los más frecuentes el medio visual y el medio sonoro. En el canal visual, el ordenador despliega una serie de objetos, que pueden ser imágenes, texto, iconos o personajes 3D que forman parte de un escenario; el usuario utiliza diversos periféricos de tipo apuntador para seleccionar los personajes de la aplicación. En el canal sonoro, el ordenador ofrece sonidos musicales, ruido de fondo o mensajes sonoros de información al usuario y éste por su parte responde de forma hablada con la información que el ordenador precisa. En este apartado se discute sobre las aplicaciones donde el uso diálogos en el canal sonoro puede aportar ventajas en la dirección de la evolución de mundos 3D, se apuntan los beneficios que pueden obtenerse, y se señalan las principales dificultades a salvar para conseguir dicho fin.

2.1. Escenarios de Aplicación

La interacción visual puede ser suficiente en multitud de escenarios y aplicaciones 3D. Cuando las necesidades de la aplicación se reducen a la navegación de personajes y a la acción sobre determinados controles u objetos, los dispositivos tipo *locator* son más que suficiente. Estas aplicaciones suelen estar orientados al entretenimiento y están basados en acciones, reflejos, y resolución de puzzles sencillos. Se caracterizan porque se apoyan en un paradigma *acción-reacción* muy simple: ante determinados eventos se producen modificaciones en el mundo de forma inmediata. Sin embargo, hay otras aplicaciones más complejas donde este paradigma puede y debe ser superado.

Así, en aplicaciones donde sea necesaria la transmisión de conceptos, de actitudes o la consecución de objetivos más desarrollados, es necesario plantear una estrategia para solucionar las acciones, que puede estar basada en la ejecución de una serie de comportamientos o tareas a realizar con varios niveles de complejidad. El usuario y los actores 3D, deben tomar decisiones no sólo tácticas sino también de tipo estratégico y con consecuencias a medio/largo plazo. En estas situaciones el contenido semántico por transacción suele ser más alto, y es aquí donde el diálogo vocal resulta más atractivo. Algunos ejemplos interesantes donde aparecen este tipo de escenarios pueden ser:

- Aplicaciones de planificación de tareas con actores 3D.
- Entrenamiento con personajes con comportamiento humano como en simulaciones de situaciones críticas o en aplicaciones de desarrollo de capacidades de liderazgo y manejo de grupos.
- Aplicaciones de enseñanza en casos especiales como la enseñanza de actitudes sociales en readaptación social o el tratamiento de discapacidades como el autismo.

La incorporación de un canal de interacción sonoro en estos escenarios puede aportar importantes beneficios. Algunos de ellos se describen en la sección siguiente.

2.2. Beneficios

Apuntamos aquí cuatro aportaciones del uso de sistemas de diálogo en las aplicaciones avanzadas arriba descritas: (1) el incremento en la naturalidad y en la capacidad inmersiva; (2) el apoyo en la planificación y realización de acciones de alto nivel; (3) el incremento en la atención del usuario y (4) la selección avanzada de objetos. El punto (1) fue discutido en la introducción y está sobradamente defendido en [5]. A continuación describimos las tres últimas.

Un agente inteligente que tenga que desarrollar un objetivo de alto nivel en un escenario 3D dado, tendrá que resolver múltiples acciones elementales cuya ejecución va a depender de la semántica de la tarea concreta y del estado del mundo. Así las

acciones de alto nivel evolucionan de forma diferente en función de diversos atributos propios del agente y/o de atributos del mundo. Los diálogos pueden servir al actor 3D o al usuario para resolver situaciones en las que se presenten alternativas que dependen del valor de diversos atributos. Mediante el diálogo se pueden introducir nuevos valores a los atributos que resuelvan ambigüedades en la ejecución de las acciones.

El uso de diálogos en estos escenarios complejos puede ayudar a centrar el foco de atención del usuario en la tarea a realizar (ver e.g. [6]). El uso de guiones (*storyboard*) desarrollados en base a diálogos va a permitir focalizar el interés del usuario y guiarle en su exploración de los problemas a resolver para realizar el objetivo concreto. Frente a otros enfoques, el uso de diálogos obliga al usuario a ser activo en el desarrollo de las tareas y permite a la máquina dirigir la conducta del mismo en el escenario (ver e.g. [7]).

Con respecto a la selección avanzada, los apuntadores deícticos son rápidos, pero su poder expresivo es limitado: normalmente sólo es posible la *selección* de un objeto. Pueden hacerse selecciones múltiples encadenando varias opciones en un menú, pero la naturalidad de la interacción se pierde, y las selecciones se suelen restringir a la zona *visible* y más cercana del mundo. Las selecciones de objetos que están fuera del campo visual o que estén alejados, pueden incluso no ser asumibles por el usuario obligando al mismo a realizar movimientos en la cámara que pueden ser complicados o tediosos (ver e.g. [8]). A través del canal vocal se pueden producir eventos para hacer selecciones múltiples de objetos que pueden estar en partes del mundo muy distantes o no accesibles y en general pueden tener implicaciones mucho más complejas por la mayor riqueza de la capacidad expresiva del lenguaje. En un hipotético ejemplo, el usuario podría pronunciar la frase: "Quiero cambiar de color todas las sillas de esta casa" y la máquina podría responderle "Qué color quieres". Esto mismo puede hacerse empleando el canal visual, pero la interacción es más tediosa y menos natural.

2.3. El Problema de la Sincronización

Incluir diálogos en un sistema de realidad virtual aporta importantes beneficios, pero también plantea problemas que tienen que ver con la integración de las diferentes tecnologías y de los diferentes niveles conceptuales de descripción de una aplicación en un mismo escenario.

Las tecnologías implicadas en los dos campos necesitan un interfaz flexible que permita la modificación del estado del mundo ante eventos de diferente naturaleza. La modificación del estado global puede implicar modificaciones en el estado del diálogo o en la apariencia del mundo. Es necesario plantear y resolver problemas de sincronización entre las actuaciones del usuario a través del sistema de navegación clásico de realidad virtual y las solicitudes y respuestas propuestas en el diálogo, donde los eventos se producen a ritmos muy diferentes y con un contenido semántico muy distinto. La figura 1 esquematiza este hecho.

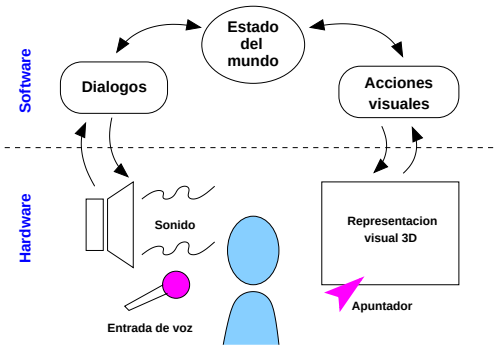

Figura 1: Canales de interacción (multimodalidad)

En las siguientes secciones se describe un marco de trabajo para desarrollar aplicaciones 3D con inclusión de diálogos y que resuelven los problemas de sincronización descritos.

3. Marco de trabajo o framework

En esta sección describimos el marco de trabajo que proponemos para el desarrollo de aplicaciones que integren mundos virtuales 3D con interacción basada en diálogos vocales. Esta plataforma puede servir para desarrollar aplicaciones basadas en mundos narrativos o guiados por diálogo. Distinguiremos dos subsistemas que se comunican entre si y que están asociados al nivel conceptual más abstracto relativo a cada canal. En la figura 2 se muestra cómo se integran los diversos componentes de nuestro marco de trabajo con una plataforma vocal/visual.

- Gestor de mundo: Debe encargarse de procesar eventos del usuario generados a través del canal visual y de mantener el estado del mundo virtual (posiciones de los agentes, animaciones, etc.).
- Gestor del diálogo: Se encargará de mantener el estado de la narración o guión, de controlar los diálogos entre el usuario y el sistema, y de proponer al gestor de mundo las acciones de alto nivel derivadas de las elecciones de usuario a través del canal vocal.

En nuestro modelo el gestor del diálogo interactúa con el usuario para determinar las acciones de alto nivel que deben llevarse a cabo. Estas acciones serán realizadas por el gestor de mundo. El estado del mismo puede determinar cómo se realiza una acción y las alternativas por las que continuará el diálogo.

Por ejemplo, ciertas acciones directas del usuario a través del canal visual pueden producir pequeños cambios de estado en el mundo. La pulsación de un interruptor

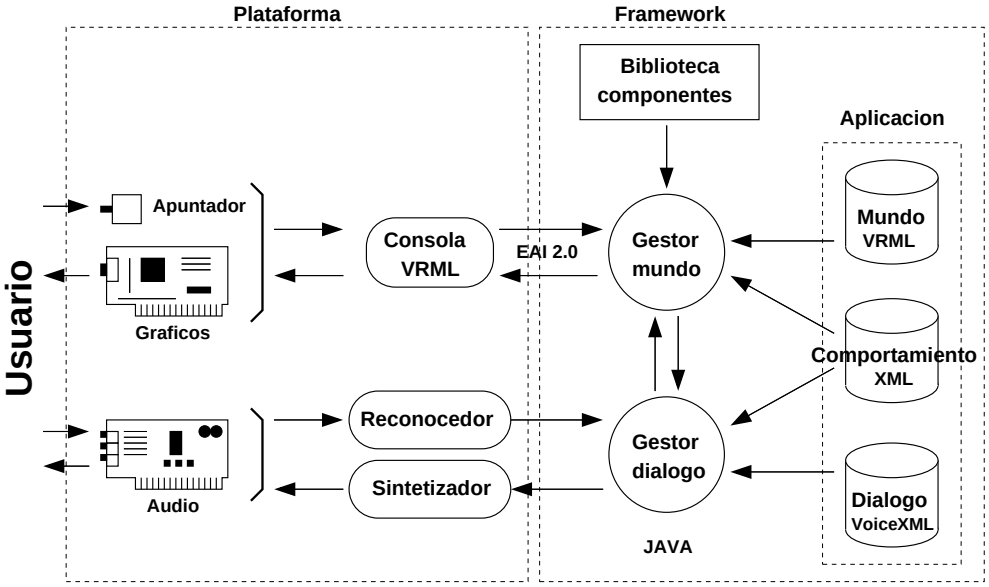

Figura 2: Esquema de integración del marco de trabajo con una plataforma

que enciende una luz no es responsabilidad del gestor del diálogo ya que este tipo de acciones no afectan directamente a la historia del diálogo. Sin embargo, al llegar a ciertas partes del guión, el hecho de que la luz esté o no encendida sí puede ser importante para su evolución.

También puede ser necesario esperar a que en el mundo se produzca una cierta situación para que el estado del guión evolucione. Por ejemplo cuando se le solicita al usuario que realice una acción directa en el mundo. El gestor del diálogo debe de poder indicar al gestor del mundo que realice acciones *asíncronas* o *sincronizadas*, de forma que el diálogo espere una respuesta del gestor del mundo relacionada con la consecución de un cierto estado.

Para construir una aplicación es preciso describir un mundo virtual y los diálogos que se llevarán a cabo con el usuario. El gestor de mundo empleará una biblioteca de componentes para realizar las acciones en el mundo virtual. Será preciso especificar como se enlazan los componentes para que se realicen las acciones y como mapean los diálogos con éstas (ver sección 4).

En las siguientes secciones describimos en detalle los dos componentes principales de nuestro framework.

3.1. El gestor del mundo

La interfaz visual muestra al usuario un mundo virtual en el que interactuar. Existirán una serie de elementos que implementan cierta funcionalidad y que permiten al usuario, además de la mera navegación por el mundo, desencadenar eventos al pulsar el ratón sobre ellos, modificando el estado del mundo.

En nuestro caso proponemos emplear VRML97 [9] como lenguaje de descripción de mundos virtuales. La principal ventaja es que se trata de un lenguaje estándar para el que existen motores de visualización e interacción disponibles (consolas VRML). La comunicación entre estos motores y un programa externo, en nuestro caso el gestor de mundo, se puede realizar a través de un interfaz de programación estándar denominado EAI [10] (*External Authoring Interface*).

El gestor de mundo es el elemento que mantiene el estado del mundo virtual. Se encargará de atender los eventos que se producen en el mismo, realizar las acciones de alto nivel y manipular la representación visual para reflejarlo. El gestor de mundo estará formado por una serie de componentes que colaboran para realizar todas estas tareas. Proponemos emplear componentes de tres niveles diferentes de abstracción: manipuladores de elementos visuales y eventos, controladores que mantiene el estado y planificadores de alto nivel (ver sección 4).

En cada aplicación concreta se definen y componen estos elementos de la forma necesaria para construir un gestor de mundo que implemente la funcionalidad requerida.

3.2. El gestor de diálogo

En el mundo virtual existirán una serie de personajes o avatares con los que, además de la interacción a través del GUI, el usuario podrá interactuar empleando voz. Dichos personajes establecerán un diálogo con el usuario, siguiendo un guión preestablecido, para la consecución de una tarea.

La interacción vocal se describe mediante VoiceXML [3] que es un lenguaje estándar para el diseño de sistemas de diálogo. Se trata de un lenguaje declarativo que nos permite centrarnos en el diseño del diálogo abstrayéndonos de la tecnología vocal utilizada: reconocedor y sintetizador de voz. Esto permite una flexibilidad total, ya que nos permite modificar los diálogos sin tener que realizar modificaciones en el resto del sistema.

El gestor de diálogo es el encargado de interpretar las páginas VoiceXML para interactuar con el usuario, empleando sintetizador y el reconocedor de voz. Una vez se ha recopilado toda la información necesaria sobre la tarea a realizar, se solicita al gestor de mundo que realice dicha tarea. A continuación, el gestor de mundo informa al gestor de diálogo sobre la situación del mundo o sobre la consecución de la tarea, de modo que se pueda seleccionar el siguiente turno de diálogo.

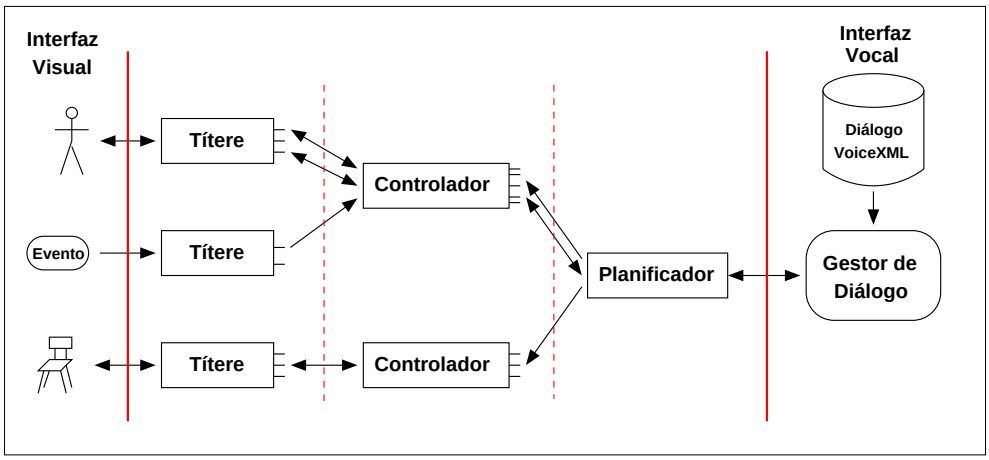

Figura 3: Modelo conceptual del gestor de mundo

4. Construcción de aplicaciones

La construcción de una aplicación concreta utilizando nuestro marco de trabajo consiste en especificar un mundo virtual, las secuencias de diálogo, las acciones que se generan y su relación con los elementos del mundo.

Como ya hemos indicado antes, las secuencias de diálogo se especifican empleando VoiceXML y los mundos virtuales utilizando VRML. Para describir el comportamiento del mundo debemos especificar: (1) la estructura de componentes que forman el gestor de mundo; y (2) la correspondencia entre los diálogos con el usuario y las acciones de alto nivel incluídas en el gestor de mundo. Toda estas especificaciones se expresarán en un documento estructurado mediante XML.

Con el objetivo de facilitar el diseño de componentes con un alto grado de reutilización, proponemos una estructuración conceptual del gestor de mundo en tres capas, ver la figura 3. Estas capas internas ponen en correspondencia las acciones desencadenadas por el diálogo con las acciones que suceden en el mundo virtual. El objetivo que perseguimos al dividir el sistema en capas es la clara separación entre las acciones de diferentes niveles de abstracción. Distinguimos tres tipos de acciones, relacionadas con los niveles conceptuales de comportamiento descritos por Roehl [11] y con el número de elementos participantes en una acción. Estos tres tipos corresponden con las tres capas que aparecen en la parte central de la figura 3 y son los siguientes:

- **Acciones elementales**: son aquellas que afectan a un único elemento del mundo virtual y que comportan una funcionalidad básica (asociadas a los niveles 0 y 1 de Roehl).
- **Acciones del mundo**: son aquellas que implican la realización de una o varias

funcionalidades elementales pero que pueden requerir la participación de varios elementos del mundo.

- **Acciones de alto nivel**: son secuencias de acciones necesarias para llevar a cabo la tarea (asociadas al nivel 2 de Roehl) desencadenada por las decisiones tomadas a través del diálogo (decisiones de nivel 3, según Roehl). Implican la utilización de una o varias acciones del mundo.

Asociaremos a cada nivel conceptual de acción un tipo diferente de componentes:

- **Títeres**: un componente de este tipo encapsula la funcionalidad de un objeto del mundo virtual. Cada función del objeto será una acción elemental, que será invocada por un controlador para que se realicen los cambios oportunos en el mundo. Por ejemplo, un títere asociado a una bombilla en el mundo virtual, puede exportar la acción *encender*.

- **Controladores**: un controlador se encarga de coordinar la invocación de las acciones elementales sobre un conjunto de títeres. Adicionalmente, el controlador mantendrá el estado del mundo, procesando en su caso los eventos del mundo virtual que lo modifiquen, que serán exportados por los títeres. Por ejemplo, un agente pulsa un interruptor y se enciende la luz. Para ello, un controlador maneja dos títeres, el del agente y el de la bombilla.

- **Planificador**: el planificador es el encargado de realizar las acciones de alto nivel. Dichas acciones pueden involucrar una secuencia de operaciones ejecutadas por uno o más controladores. Las acciones resultantes del diálogo con el usuario se mapean directamente con las acciones que permite realizar el planificador. Por ejemplo, al indicar a un agente que abandone una habitación, el planificador indicará una secuencia de acciones a un controlador para que el agente se dirija a la puerta, apague la luz, salga y la cierre.

Existen otras propuestas de especificación de comportamiento en mundos 3D utilizando XML (ver e.g. [12]). Estas se centran en la especificación declarativa de comportamientos incrustados dentro de la propia jerarquía de objetos del mundo, y su encadenamiento para realizar acciones de nivel 2. En cambio, nuestra propuesta está orientada a la especificación de un motor externo del comportamiento (gestor de mundo) organizado a partir de componentes procedimentales. Nuestra aproximación introduce en el nivel de acciones del mundo el concepto de comportamientos agrupados (en un controlador que maneja varios títeres) y propone una mejor y más clara división en capas conceptuales. Esta división facilita el enlace con un nivel de abstracción superior representado por los dialogos. Las acciones provocadas por el diálogo externo se descomponen de forma natural hasta llegar a los eventos de más bajo nivel en el mundo virtual.

5. Caso de estudio

Para demostrar las posibilidades de nuestra arquitectura hemos desarrollado un pequeño prototipo que implementa la separación del gestor de mundo y de diálogo. En la figura 5 puede verse un escenario de ejemplo para una aplicación sencilla desarrollada sobre este prototipo. El mundo consiste en un plano limitado en el que hay una serie de losas de colores, y contiene un agente representado visualmente por un pequeño robot. El objetivo del robot es desplazarse de una losa a otra trazando las rutas que desee el usuario. A través del diálogo podemos indicar al robot la ruta que deseamos de una forma natural (ver un ejemplo de interacción en la figura 4).

Sistema: ¿Qué acción debo realizar?
Usuario: Mover.
Sistema: ¿A la casilla de qué color?
Usuario: Azul.
Sistema: ¿Debo pasar por alguna casilla intermedia?
Usuario: Sí.
Sistema: ¿De qué color es esa casilla?
Usuario: Verde.

Figura 4: Ejemplo de interacción

Cuando utilizamos una cámara fija sobre la escena puede resultar más rápido utilizar el apuntador para seleccionar las losas en el orden deseado. Sin embargo, cuando utilizamos una visión subjetiva libre, o la cámara sigue al robot, no todas las losas son visibles en un instante dado (ver figura 6). Reorientarse para pulsarlas es en general más costoso que indicar de forma verbal el siguiente objetivo, empleando el paradigma de navegación por consulta (ver e.g. [13]).

Para implementar este prototipo hemos montado una plataforma de prueba en entorno PC. Utilizamos un sistema operativo Windows 2000. El interfaz visual está manejado por una consola de VRML: Cortona versión 2.4 (de libre disposición para uso personal). Esta consola se integra como un plug-in en Microsoft Internet Explorer. El gestor de mundo se ejecuta como un applet en la máquina virtual Java de Microsoft del Internet Explorer, y se comunica con Cortona a través de su implementación del EAI. Por otro lado, se comunica con el gestor de diálogo a través de una conexión de red. El gestor de diálogo es un programa Java que incluye un interprete VoiceXML y controla la plataforma vocal, implementada mediante los motores de síntesis y reconocimiento desarrollados por la Universidad Politécnica de Cataluña y distribuidos por la empresa ATLAS.

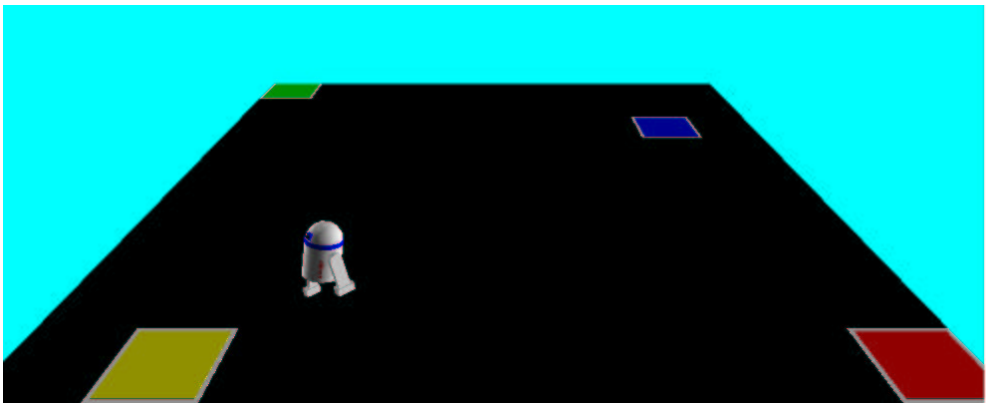

Figura 5: Mundo virtual desde cámara fija

6. Conclusiones

La incorporación de diálogos hablados en aplicaciones de inmersión en mundos virtuales contribuye claramente a incrementar la naturalidad y posibilidades de la interacción, sirve de apoyo en la planificación y realización de acciones complejas de alto nivel de abstracción, contribuye a aumentar la atención del usuario en las actividades sensomotoras que complementan las acciones y facilita la selección avanzada de objetos de interés.

En este trabajo se ha descrito el marco de desarrollo de aplicaciones que proponemos para facilitar la integración sistemática del canal hablado en la interacción con escenarios de realidad virtual. Aprovechando nuestra experiencia en desarrollo de sistemas de diálogo basados en el estándar de especificación VoiceXML hemos mostrado que resulta sencillo incorporar guiones a mundos que pueden estar especificados también en alto nivel en XML. El estándar EAI para VRML ha demostrado, más allá de las limitaciones tecnológicas vinculadas a las restricciones que impone al uso de ciertas plataformas, ser un buen punto de partida para la implementación de la comunicación entre el subsistema gestor del mundo virtual y el gestor de diálogo.

Los resultados presentados aquí suponen un punto de partida hacia la integración de otras recomendaciones de especificación en XML de la apariencia y el comportamiento en mundos virtuales con las de especificación de la interacción con el usuario basada en diálogo hablado natural que defendemos.

Referencias

[1] J. Gusfafson, L. Bell, J. Beskow. AdApt - a multimodal onversational dialogue system in an apartment domain. Proc of ICSLP 2000

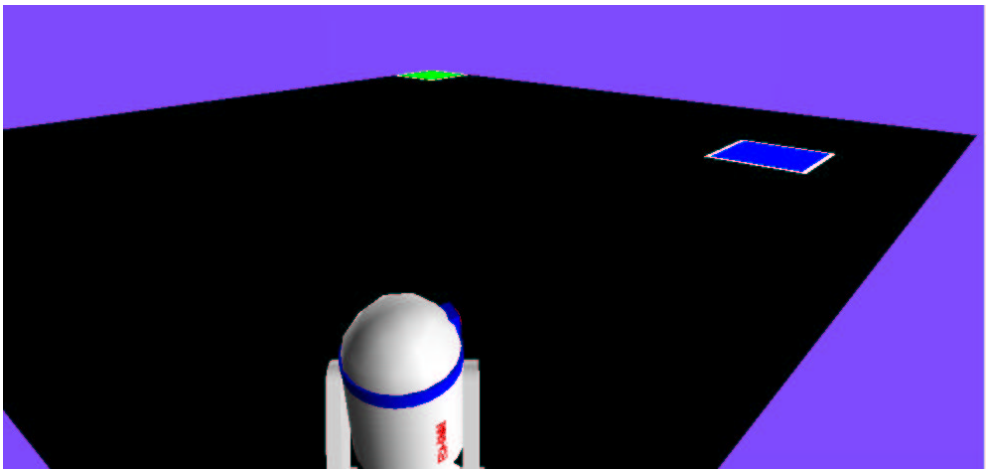

Figura 6: Mundo virtual desde cámara móvil

[2] P. Nugues. Verbal Interaction in Virtual Worlds. In proc. CHI 2000 workshop on Natural Language Interfaces

[3] Voice eXtensible Markup Language (VoiceXML) Version 2.0. Proposed Recommendation 3 February 2004 W3C Voice Browser Working Group. <http://www.w3.org/TR/2004/PR-voicexml20-20040203/>.

[4] D. Traum and J. Rickel Embodied Agents for Multi-party Dialogue in Immersive Virtual Worlds. In Proceedings of the 1st International Joint Conference on Autonomous Agents an d Multi-Agent Systems (AAMAS 2002).

[5] S. McGlashan and T. Axling Talking to Agents in Virtual Worlds Proc. 3 UK Conference on Virtual Reality Leicester, England, 1996

[6] W. Swartout, M. van Lent. Making a Game of System Design. Communications of the ACM, Vol. 46(7), July 2003.

[7] N. Gershon, W. Page. What Storytelling can do for Information Visualization. Communications of the ACM, Vol. 44(8), August 2001.

[8] E. Kaiser, A. Olwal, D. McGee, H. Benko, A. Corradini, X. Li, P. Cohen, S. Feiner. Mutual Disambiguation of 3D Multimodal Interaction in Augmented and Vitual Reality. ICMI PUI 03 Vancouver, Canadá

[9] VRML97 The Virtual Reality Modeling Language - International Standard ISO/IEC 147772-1:1997. The VRML Consortium Inc., 1997.

[10] The Virtual Reality Modeling Language (VRML) – Part 2: External authoring interface (EAI). International Standard ISO/IEC FDIS 14772-2:2002 The VRML Consortium Inc., 2002.

[11] B. Roehl. Some Thoughts on Behavior in VR Systems. 1995 <http://ece.waterloo.ca/~broehl/behav.html>.

[12] R. Dachselt, E. Rukzio. Behavior3D: An XML-Based Framework for 3D Graphics Behavior. In proc. eighth international conference on 3D web technology. ACM, 2003

[13] A. van Ballegooij, A. Eliëns. Navigation by Query in Virtual Worlds. In proc. WEB3D'2001. ACM Press, 2001.