ARQUITECTURA MULTIMODAL CONTROLADA POR VOZ: REVISIÓN DE METÁFORAS DE INTERACCIÓN

David Escudero-Mancebo, Héctor Olmedo-Rodríguez, Valentín Cardeñoso-Payo
ECA-SIMM Laboratory, Universidad de Valladolid[1]
Campus Miguel Delibes s/n. 47008 Valladolid
{descuder, holmedo, valen}@infor.uva.es

RESUMEN

En esta comunicación presentamos una plataforma para el desarrollo de aplicaciones multimodales dirigidas por voz. Se presenta un lenguaje de especificación de escenas multimodales y la arquitectura soporte. El lenguaje y la arquitectura se validan revisando la cobertura de una serie de metáforas básicas que tienen que ver con la interacción gráfica, interacción vocal y la combinación de ambos modos. Esta evaluación pone en evidencia las capacidades de la plataforma y apunta el trabajo futuro que debe realizarse.

1. INTRODUCCIÓN

Una de las aplicaciones más interesantes de las tecnologías del habla (TH) es su uso para implementar interfaces multimodales [1]. La voz puede ser un complemento a la interacción gráfica en el uso de determinados terminales tipo kiosco o en terminales móviles. Otro ámbito donde las TH pueden ser un complemento importante es el de las aplicaciones de entretenimiento o de entrenamiento (AEE). Uno de los ámbitos más utilizados para el desarrollo de aplicaciones de (AEE) son los entornos 3D.

A pesar del espectacular crecimiento tanto de las aplicaciones 3D como de la investigación en TH el estado del arte en aplicaciones multimodales que combinen 3D con sistemas de diálogo se caracteriza por la presencia de un número pequeño de prototipos. Aunque hay propuestas de estandarización, los prototipos existentes parecen soluciones ad-hoc en un ámbito reducido y con aplicación limitada (con excepción quizá de las aplicaciones de visual speech). En este artículo presentamos una plataforma para el desarrollo de aplicaciones 3D que incluyan interacción vocal cuyo objetivo es ofrecer un marco genérico para la programación conjunta de escenas y personajes 3D y de diálogos teniendo en cuenta los estándares disponibles en ambos ámbitos.

La plataforma es descrita en más detalle en [2]. En esta comunicación se plantea además una propuesta para la evaluación de la plataforma propuesta. Esta evaluación se realiza en base a un análisis de cobertura de una serie de metáforas de interacción básicas encontradas en la bibliografía básica. Este análisis permite destacar los méritos de la plataforma y apuntar los aspectos donde debe ser mejorada.

Primero presentaremos nuestra propuesta de interacción multimodal (gráfica y vocal) con espacios 3D. Seguidamente enumeraremos las metáforas de interacción gráfica y vocal así como los tipos de cooperación entre éstas a la vez que revisamos las posibilidades de la plataforma para dar cobertura a dichas metáforas. Las conclusiones destacan el trabajo futuro a realizar para mejorar la plataforma.

2. LA PLATAFORMA MULTIMODAL

En este apartado describiremos el lenguaje de especificación y la arquitectura propuesta para definir aplicaciones que permiten interactuar de manera multimodal con entornos 3D.

2.1. El lenguaje XMMVR

El eXtensible markup language for MultiModal interaction with Virtual Reality worlds o XMMVR es un lenguaje de marcas para especificar escenas, comportamientos e interacción. Cada mundo es modelado por un elemento XMMVR, usando la metáfora de película cinematográfica. Es un lenguaje de marcas híbrido porque en éste quedan embebidos otros lenguajes como VoiceXML o X+V para interacción vocal y X3D o VRML para describir las escenas 3D. Los ficheros XML válidos para el DTD de XMMVR incluyen enlaces a los programas y ficheros necesarios para ejecutar el mundo 3D definido. Nuestro sistema es dirigido por eventos, por ello se requiere definir una mínima lista de eventos para sustituir la línea de tiempo.

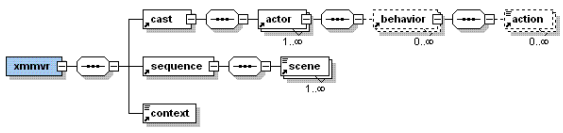

Fig. 1. *Elementos del lenguaje XMMVR*

[1] Trabajo parcialmente financiado por el proyecto de la Junta de Castilla y León (VA077A08)

La Figura 1 muestra la estructura de un documento XMMVR. Cualquier elemento *xmmvr* está formado por un reparto de actores llamado *cast* y una secuencia de escenas llamada *sequence* determinando la evolución temporal del mundo. El elemento *context* se reserva para uso futuro.

Al usuario se le considera un miembro de la audiencia. Es capaz de interactuar con los actores del mundo aunque no es considerado como un actor del mundo.

Cada *actor* del reparto es un elemento con apariencia gráfica descrita en un fichero VRML y un comportamiento *behaviour* que especifica la interacción del usuario. Cada comportamiento está definido como un par <*evento, lista de acciones*>. Las acciones son ejecutadas cuando una condición *condition* se cumple.

El usuario genera eventos utilizando la interacción gráfica *GUI* o la interacción vocal *VUI*. Existen también eventos del sistema para definir la interacción con otros actores del mundo (*eventos ACT*) o para interactuar con el sistema (eventos *SYS*). La *lista de acciones* es un conjunto de acciones a ejecutarse cuando ocurre un evento. Las acciones pueden ser de tipo GUI, VUI, ACT o SYS. Las acciones GUI modifican la apariencia gráfica del mundo 3D. Las acciones VUI son diálogos. Las acciones ACT son mensajes enviados entre actores. Acciones SYS son navegaciones entre escenas.

El elemento *sequence* planifica las escenas *scenes* del mundo. Por defecto las escenas se muestran en el mismo orden en el que están escritas en el documento. Los eventos y acciones SYS permiten navegar entre escenas cambiando el orden secuencial por defecto. Debe definirse por lo menos una escena en el mundo xmmvr. La interacción es sólo posible si se ha definido al menos un actor.

De acuerdo a estas premisas, se ha definido un DTD [3] de manera que es posible desarrollar aplicaciones multimodales escribiendo un fichero XML válido para el DTD de XMMVR. Las escenas 3D, los actores, sus comportamientos y la interacción con el usuario se definen en el mismo fichero de marcas. Este fichero se usa por la arquitectura del sistema para ejecutar la aplicación como se describirá a continuación.

2.2. La plataforma XMMVR

Hemos creado un marco para desarrollar aplicaciones de interacción persona-ordenador multimodales donde el flujo de la interacción en entornos 3D está conducido por diálogos hablados. Para construir una aplicación utilizando nuestro marco el desarrollador tiene que especificar el mundo virtual, la secuencia de diálogos y la lista de acciones a lanzarse cuando los eventos son generados por los usuarios. En la sección anterior hemos descrito un lenguaje para especificar estos elementos en un documento XMMVR común. En esta sección describimos la arquitectura del sistema responsable de analizar sintácticamente documentos XMMVR y ejecutar la correspondiente aplicación multimodal.

Una de nuestras metas fue construir una aplicación web multimodal, por lo que desarrollamos un sistema embebido en un navegador web. Los diálogos están programados utilizando VoiceXML y las escenas 3D y actores están descritos en VRML. Hemos desarrollado un gestor de mundo para planificar las acciones a ejecutarse en el mundo 3D utilizando un applet Java.

2.2.1. Gestión del diálogo vocal

La Figura 2 muestra los componentes que gestionan la interacción vocal. Existe un componente embebido en el navegador web que inicia la ejecución solicitando un diálogo al dispatcher. Recupera el documento VXML correspondiente de un repositorio y lo envía al intérprete. El intérprete VXML ejecuta el diálogo utilizando la plataforma vocal. El interfaz entre el intérprete VXML y la plataforma vocal son prompts (a ser sintetizados) y fields (la salida del reconocedor de diálogo automático) de acuerdo al estándar VoiceXML.

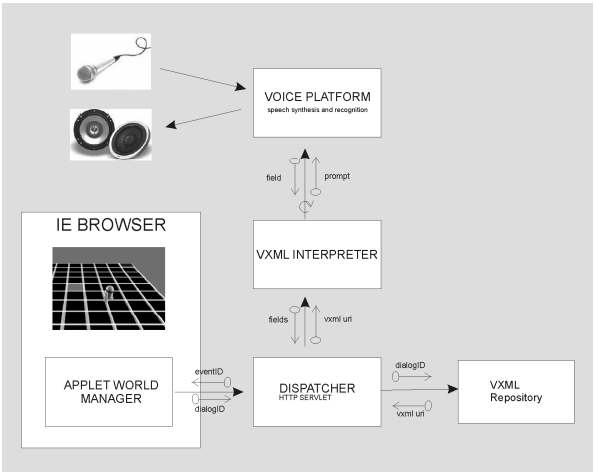

Fig. 2. *Componentes de la arquitectura para la gestión del diálogo hablado*

Hemos desarrollado un sistema que utiliza un applet Java en un navegador web Internet Explorer. Utiliza el navegador VRML CORTONA [4] para mostrar el estado del mundo. La interacción GUI se basa en el API EAI [5]. Utilizamos un servlet sobre un servidor Apache Tomcat para alimentar el navegador vocal de nuestro sistema de diálogo. Creamos nuestro propio intérprete VXML que utiliza la plataforma vocal ATLAS de Ibervox [6]. Como los componentes vocales están distribuidos sobre diferentes servidores, las aplicaciones multimodales se pueden ejecutar en un PC convencional con audio y los navegadores apropiados.

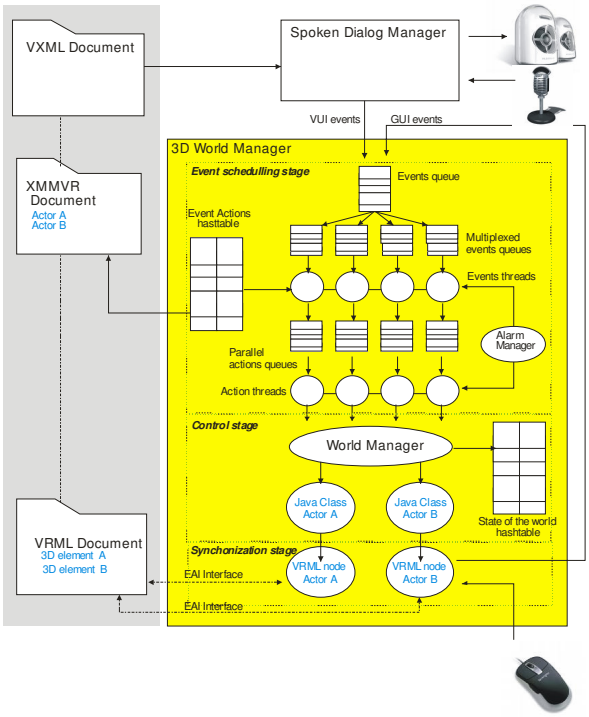

Fig. 3. *Componentes de la arquitectura del gestor del mundo 3D*

2.2.2. Gestión del mundo 3D

Los componentes necesarios para hacer al mundo 3D reaccionar a eventos se muestran en la Figura 3. Los eventos VUI se despachan por el gestor de diálogo hablado y los eventos GUI se generan típicamente clickando los elementos 3D. El sistema obtiene la información acerca de las escenas 3D y los actores del documento XMMVR. El mismo documento especifica el comportamiento de los actores como pares <*evento, lista de acciones*> y enlaza con el documento VRML que especifica la escena 3D y los actores. La parte izquierda de la Figura 3 muestra la relación entre dos actores XMMVR llamados actor A y actor B y dos elementos 3D VRML llamados nodo A y nodo B. En respuesta a eventos, el sistema hace cambios en los nodos de acuerdo a especificaciones del documento XMMVR.

La primera etapa del sistema es la tarea de Gestión de eventos, donde los eventos son encolados y multiplexados en un número de colas paralelas (en este caso cuatro colas, pero este parámetro es programable). Cada cola es atendida por el hilo correspondiente, y esto permite la ejecución concurrente de acciones en la escena. Los hilos acceden a la tabla hash que contiene las acciones correspondientes con los eventos, como se describe en el documento XMMVR. Si alguna de las acciones no es elemental, el hilo se encarga de descomponerla, produciendo una nueva cola de acciones elementales preparada para ser ejecutada por el Gestor del mundo en el próxima etapa. Hemos programado un Gestor de alarmas para añadir anti-acciones que eliminan varias acciones a ser canceladas cuando ocurren eventos excepcionales (alarmas).

Cada elemento 3D VRML tiene dos clases Java asociadas: Control e Interfaz. Control ejecuta acciones elementales e interacciona con las variables correspondientes de la tabla hash "Estado del mundo" e Interfaz manipula los gráficos del nodo 3D. Entre la clase Java Interfaz y los nodos 3D hay un interfaz (escenario *synchronization* en la Figura 3) que es responsable de utilizar el EAI para manipular los elementos VRML y recibir los eventos GUI y direccionarlos a la entrada del sistema. En la Figura 3, el nodo A es manipulado a través de la clase Java Interfaz Actor A. La diferencia entre el nodo A y el nodo B es que el nodo B puede potencialmente enviar eventos GUI tales como clicks de ratón en el nodo 3D VRML B.

El Gestor del mundo lee las acciones elementales y llama a los métodos Java correspondientes. Además, actualiza la tabla hash "Estado del mundo" que almacena el estado de las propiedades del mundo. La información de esta tabla hash es utilizada para asignar valores a los parámetros de los métodos Java. El gestor es también responsable de actualizar esta tabla de acuerdo a las acciones ejecutadas.

3. REVISIÓN DE METÁFORAS

En este apartado presentaremos las características y clasificación de las metáforas de interacción gráfica y vocal así como los tipos de cooperación entre ellas para evaluar las capacidades de nuestra propuesta de cara a su resolución.

3.1. Interacción gráfica

Empleamos las referencias [7] y [8] que clasifican los diferentes modos de interacción gráfica de acuerdo a metáforas fundamentales (teatro y locomoción) y metáforas navegacionales (elevador, vehículo, deslizamiento, silla voladora y tele transporte). Se ha comprobado que el lenguaje XMMVR tiene expresividad suficiente para atender estas metáforas. No entramos en detalle en este artículo porque estamos más interesados aquí en la interacción vocal.

3.2. Interacción vocal

Oviatt [9] distinguió, entre las características básicas para la interacción multimodal, la necesidad de restringir el lenguaje por las limitaciones de los sistemas de diálogo para entender el lenguaje natural y la necesidad de realimentación para adaptar el lenguaje a nuevas situaciones o cambios de estado.

El uso de VXML impone diálogos restringidos y permite la realimentación.

McGlashan [10] distingue cuatro metáforas de interacción vocal:

Proxy o Delegado: El usuario puede tomar control de varios agentes (cambiar de agente, selección en la acción) en el mundo virtual e interactuar con el mundo virtual a través de ellos, por ejemplo: pintor, ¡pinta la casa de rojo!
Divinity: El usuario actúa como un dios y controla el mundo directamente, por ejemplo: que la casa sea roja!
Telekinesis: Los objetos y agentes en el mundo virtual pueden ser interlocutores de diálogo del usuario, por ejemplo: casa, ¡píntate de rojo!
Interface Agent: El usuario se comunica con un agente, separado del mundo virtual, que ejecuta sus comandos.

El sistema está preparado para resolver las metáforas de *Proxy* e *Interface Agent* ya que los diálogos están vinculados vía el elemento *behavior* a los distintos actores. Sin embargo, el uso de la metáfora *Proxy* exigiría la disponibilidad de un intérprete VXML que dispusiera de diferentes voces para asociar a los diferentes agentes que fueran a intervenir.

Las metáforas *Telekinesis* y *Divinity* podrían simularse empleando un actor sin apariencia gráfica que fuera responsable de dialogar con el usuario e interactuar con el resto de agentes vía eventos de tipo ACT.

3.3. Cooperación entre modalidades

Existen cinco tipos básicos de cooperación entre modalidades según [11]:
Transferencia: Parte de la información producida por una modalidad es usada por otra modalidad.
Equivalencia: Ambos modos podrían tratar la misma información.
Especialización: Un determinado tipo de información es siempre procesada por la misma modalidad.
Redundancia: La misma información es procesada por ambas modalidad.
Complementariedad: Diferentes partes de información son procesadas por cada modalidad pero tienen que ser combinadas.

La *transferencia* está garantizada por el mantenimiento de un contexto tanto en el lenguaje XMMVR como en el *Gestor del mundo* de la arquitectura (*hash* de Estado del mundo). Uno de los modos puede provocar la ejecución de una acción que modifique el valor de estas variables de contexto y esta información ser usada por el otro modo.

La *complementariedad* también puede ser programada usando el contexto en el lenguaje XMMVR.

La *equivalencia* también puede darse ya que la misma secuencia de acciones puede ser ejecutada como consecuencia de un evento vocal o gráfico. La *especialización* es responsabilidad del programador de escenas que puede delegar determinadas tareas a uno u otro modo.

Por último, la *redundancia* es el caso más problemático. Cuando los dos modos introducen información que se refiera a lo mismo pueden darse situaciones de bloqueo dado que las acciones pueden ejecutarse de forma concurrente. Esto está previsto con la inclusión del elemento *Alarm manager* que se espera pueda anular acciones de las colas. Sin embargo existen situaciones que van a precisar modificaciones en la arquitectura como pueden ser la cancelación de diálogos cuando el modo gráfico introduzca una información redundante.

5. CONCLUSIONES Y TRABAJO FUTURO

Hemos presentado nuestra propuesta para definir e implementar aplicaciones que permitan interacción multimodal con entornos 3D. Hemos revisado el estado del arte referente a metáforas de interacción gráfica y vocal así como los tipos de cooperación entre distintas modalidades. Tras ello, hemos evaluado nuestra propuesta concluyendo que nos permite implementar de manera notable metáforas de interacción gráfica estructurales y navegacionales. Se hace más difícil la implementación de las distintas metáforas de interacción vocales así como los tipos de cooperación entre modalidades.

De estas deficiencias que detectamos en nuestro sistema podemos definir nuestro trabajo futuro que podemos resumir en: adaptar la solución propuesta a resolver cada metáfora de interacción gráfica y vocal y cada tipo de cooperación entre distintas modalidades.

6. BIBLIOGRAFÍA

[1] Jaimes A., Sebe N. Multimodal human-computer interaction: A survey**.** IEEE IW on HCI & ICCV 2005.
[2] Olmedo H. et al. Conceptual and practical framework for the integration of multimodal interaction in 3D worlds, New Trends on HCI. Springer (en prensa).
[3] DTD de XMMVR. http://www.xmmvr.info/.
[4] Cortona: http://www.parallelgraphics.com/.
[5] Phelps A.M. Introduction to the External Authoring Interface, EAI. Rochester Inst. of Technology, Dep. of Information Technology, 1999.
[6] Atlas Ibervox: http://www.verbio.com/.
[7] Contigra. http://www.contigra.com/.
[8] Dachselt R., Action Spaces - A metaphorical concept to support navigation and interaction in 3D interfaces; User Guidance in Virtual Environments. Workshop "Usability Centred Design and Evaluation of Virtual 3D Environments", 2000.
[9] Oviatt S., Cohen P., Multimodal interfaces that process what comes naturally. Com. of the ACM, 2000.
[10] McGlashan S., Axling T., Talking to Agents in Virtual Worlds. UK VR-SIG Conference, 1996.
[11] Martin, J. C. TYCOON: theoretical and software tools for multimodal interfaces, AAAI Press., 1998.