Propuesta de Especificación y Arquitectura para aplicaciones de Interacción Multimodal en Escenarios 3D

H. Olmedo, D. Escudero, A. González, C. González, V. Cardeñoso
ECA-SIMM, Departamento de Informática. E.T.S.I.I. Universidad de Valladolid
Campus Miguel Delibes s/n. 47011 - VALLADOLID
holmedo@infor.uva.es

Resumen

La definición de un lenguaje de marcas para modelar escenas, comportamiento e interacción en base a la metáfora de película cinematográfica interactiva puede servir de marco común para desarrollar aplicaciones que permitan interacción multimodal en escenarios 3D. Presentamos las bases de una arquitectura que nos permita integrar los componentes de este tipo de aplicaciones de interacción multimodal en entornos virtuales 3D reutilizando lenguajes de marcas ya definidos.

Palabras clave: Entornos virtuales 3D, comportamiento, interacción vocal, interacción gráfica, interacción persona-ordenador

1. Introducción

Los sistemas basados en Entornos Virtuales 3D (EV3D) incrementan significativamente el potencial de Interacción Hombre Máquina [1] y los Sistemas de Diálogo (SD) aportan un canal complementario al canal gráfico como es el canal vocal o sonoro [2]. Por ello, la integración de ambos campos de investigación debería ser una evolución natural de ambas tecnologías, pero no ha sido apenas explotada en sistemas comerciales. Aunque existen prototipos [3], se trata de un ámbito de trabajo por descubrir. La principal razón por la cual no existen apenas soluciones integradoras de EV3D y SD, es la juventud de estas áreas de trabajo, donde la mayoría de los esfuerzos se han centrado en mejorar de forma separada ambos campos, y no en estudiar las necesidades de interdependencia que se derivan de una propuesta integradora. Aquí se presenta una propuesta que combina EV3D-SD planteando una plataforma para desarrollo de aplicaciones basadas en mundos virtuales 3D que permita una interacción multimodal dirigida por diálogos.

Los ámbitos de los EV3D y de los SD se caracterizan por una relativa disponibilidad de prototipos y sistemas comerciales que, por lo general, han descuidado la necesidad de ajustarse a algún estándar de desarrollo o de especificación. El estándar en SD es VoiceXML [4] mientras que en EV3D hay un estándar de definición de escenas X3D [5] evolucionado de VRML [6]. Estos estándares han supuesto un marco de referencia para que los desarrolladores adapten sus sistemas, con las consiguientes aportaciones en cuanto a facilidad de uso en lo que se refiere a la definición de escenarios 3D y diálogos y a la portabilidad de módulos reutilizables. Presentamos un marco de referencia que pretende ser un lenguaje de especificación de mundos 3D con integración de diálogos. Respeta los estándares disponibles para EV3D y SD, sirve de vínculo entre ambos mundos y aporta una coherencia argumental.
En este artículo introduciremos la interacción gráfica, la vocal y la problemática de su fusión. Describiremos el lenguaje XMMVR definido. Presentaremos la arquitectura necesaria para poder implementar este tipo de aplicaciones, las bases en las que hemos centrado su diseño y los elementos utilizados. Finalizaremos con las conclusiones y el trabajo a realizar en el futuro.

2. Interacción multimodal 3D y metáforas

Añadir interacción vocal a los EV3D aporta beneficios tales como emitir comandos manteniendo la libertad de manos y ojos. Los usuarios pueden referirse a objetos que no están presentes en la vista actual del mundo virtual, lo que hace que las acciones sean rápidas y su efecto inmediato. Pero existe una dificultad para la aproximación general a fusión multimodal que hace necesaria la definición de una arquitectura reutilizable para construir nuevos sistemas multimodales. Las tres componentes de la

interacción multimodal para EV3D son: la especificación tridimensional que básicamente consiste en modelar objetos del entorno virtual que pueden ser estáticos y/o dinámicos; la interacción gráfica (GUI) basada en teclado y ratón como la conocemos hasta ahora y que siempre gira en torno al modelo de eventos y en base a espacios de acción o action spaces [7] que son aproximaciones metafóricas para estructurar los interfaces de usuario tridimensionales; y por último, la interacción vocal (VUI) en la que son posibles cuatro metáforas de interacción [8]. Elegir la metáfora de interacción vocal adecuada a nuestro mundo simplificará la especificación de un lenguaje que englobe ésta dentro del marco definido.

3. El lenguaje de especificación propuesto

El eXtensible markup language for MultiModal interaction with Virtual Reality worlds o XMMVR es una propuesta de definición de un lenguaje de marcas para definir escena, comportamiento e interacción en el que consideraremos cada mundo o película interactiva como un elemento *"xmmvr"* basándonos en la metáfora de película cinematográfica. Es un lenguaje de marcas híbrido porque utiliza otros lenguajes como VoiceXML para interacción vocal y X3D o VRML para descripción de escena. El procesamiento de los ficheros XML válidos para el DTD de XMMVR permitirá enlazar con los programas y ficheros necesarios para hacer funcionar el mundo especificado. Nuestro sistema será dirigido por eventos, por ello habrá que definir una mínima lista de eventos. Un elemento *"xmmvr"* está formado principalmente por el reparto de actores *"cast"* y la secuencia de escenas *"sequence"* que marcan el transcurrir del mundo. El elemento *"cast"* o reparto será el conjunto de actores que intervendrán en el mundo o película *"xmmvr"* es decir, cada uno de los elementos que tienen una apariencia gráfica especificada por un fichero VRML y un comportamiento que permite una interacción con el usuario. Al usuario lo consideraremos como un espectador sin presencia en el mundo pero que interactúa con los actores de éste, por tanto estamos utilizando la metáfora del proxy o delegado para especificar la interacción vocal y puesto que nos basamos en la metáfora de la película cinematográfica interactiva, ésta sería una evolución de una metáfora fundamental de interacción gráfica: la metáfora de teatro.

Diremos que un *"actor"* es todo elemento que puede formar parte del mundo definido y que tendrá comportamientos propios que especificaremos con la etiqueta *"behavior"*.

Cada comportamiento *"behavior"* se definirá como una pareja de *evento* y *lista de acciones* que puede tener cada actor ante una determinada *condición*. Un evento puede ser provocado por el usuario debido a una interacción gráfica *"GUI"* o a una interacción vocal *"VUI"*. Asimismo existen eventos del sistema que sirven para definir la interacción con otro actor del mundo *"ACT"* o de interacción con el mundo o sistema *"SYS"*. La lista de acciones serán una o varias acciones que se generan ante un evento y pueden ser también de carácter gráfico *"GUI"*, vocal *"VUI"*, de interacción con otro actor del mundo *"ACT"* o de interacción con el mundo o sistema *"SYS"*.

Tenemos que especificar también la secuencia de escenas *"sequence"* en la que tendremos una o más escenas que se presentarán por defecto en el orden en el que se escribieron. Consideraremos que ocurre al menos una escena *"scene"* y para que haya interacción entre el usuario y esa escena del mundo, deberá existir al menos un *"actor"* que habite el mundo definido. Con todas estas premisas hemos definido un DTD [9] y podremos desarrollar aplicaciones para interacción multimodal con mundos virtuales en base a archivos XML válidos para el DTD / XML Schema que representamos en la figura 1.

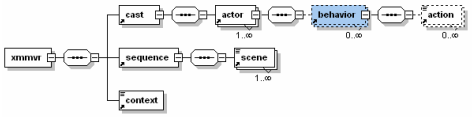

Figura 1. XML Schema de XMMVR

4. La plataforma propuesta

Queremos disponer de un marco y un lenguaje para modelar aplicaciones de interacción hombre-máquina multimodales (interacción gráfica e interacción vocal) en EV3D. Así la construcción de una aplicación concreta consistirá en especificar un mundo virtual, las secuencias de diálogo, las acciones que se generan y su relación

con los elementos del mundo. Las secuencias de diálogo se especifican empleando VoiceXML y los mundos virtuales utilizando VRML. Para describir el comportamiento del mundo debemos especificar la estructura de componentes que lo forman, la correspondencia entre los diálogos con el usuario y las acciones de alto nivel en un documento estructurado mediante XML conforme al DTD de XMMVR. Desarrollaremos así una aplicación "embebida" en un navegador web que permita a un usuario controlar un mundo virtual a través del micrófono y del teclado/ratón del ordenador donde se está ejecutando. Este applet sobre un navegador HTML permite a nuestro navegador VRML mostrar el estado del mundo sobre el que interactuamos. Para ello hemos desarrollado varios paquetes Java basados en el API EAI pero integrando desarrollos anteriores realizados por el grupo ECA-SIMM. Con la ayuda de un servlet sobre un servidor Apache Tomcat realimentaremos el navegador vocal de nuestro sistema de diálogo ATLAS de IBERVOX [10]. Todo esto se ejecutará sobre un PC normal con un micrófono y unos altavoces conectados a su tarjeta de sonido.

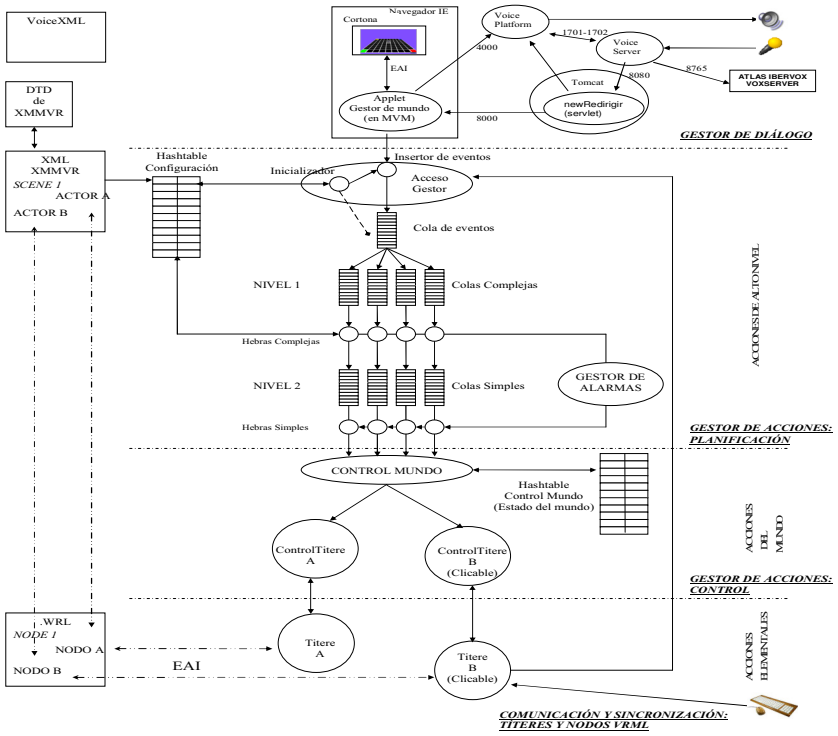

Figura 2. Arquitectura XMMVR

Nuestra plataforma de control de flujo se divide en varias capas superpuestas donde cada una da un servicio a la capa inmediatamente superior. Seguiremos un orden de arriba hacia abajo en el nivel de abstracción según la figura 2:

- Capa superior: Gestor de diálogo que genera eventos procedentes de una interacción vocal con el usuario y los proporciona como entrada para la capa inmediatamente inferior.
- Gestor de acciones: Subsistema que recibe eventos (vocales y gráficos), los traduce en series de acciones y planifica su ejecución. Se subdivide en dos capas:
  - Planificación: Provee un mecanismo de acceso unificado para la entrada de eventos de diferentes fuentes. Una vez dentro, estos eventos se traducen en unas acciones complejas y cada una de éstas se subdivide en una serie de acciones

simples. Dichas series de acciones pueden ser ejecutadas de forma paralela pero las acciones simples procedentes de una misma acción compleja se ejecutan de forma secuencial. Además esta capa implementa una arquitectura para el tratamiento de "anti-acciones" que son mandatos que anulan acciones simples pendientes de procesarse.

- Control: Recibe las acciones simples procedentes del nivel superior. Aquí se ejecutan estas acciones teniendo en cuenta el estado del mundo (conjunto formado por los estados de todos los elementos dinámicos que componen el mundo virtual). Tras la ejecución de una acción simple el estado del mundo se actualizará con los nuevos estados.
- Comunicación y sincronización: Base para ejecutar las acciones sobre el mundo virtual con un nivel superior de abstracción. Esto nos evita tener que conocer la interfaz EAI mediante la cual Java se comunica con un mundo en VRML y nos provee de un mecanismo de sincronización para la interacción con los objetos del mundo. Esta capa admite la posibilidad de que un elemento virtual genere eventos, al pinchar con el ratón sobre su representación visual en el mundo (títere clickable). Se considera que un *Títere* se compone de dos naturalezas separadas: Por una parte es un objeto tridimensional desarrollado en el lenguaje de modelado, *Nodo VRML*. Este objeto tan solo es una imagen sin ningún tipo de comportamiento o capacidad de realizar acciones. Sería la *marioneta* de un títere. Por otro lado es una *clase* escrita en Java a la que está asociado el objeto del mundo virtual y que es la encargada de proporcionar funcionalidad y dotar de la capacidad de realizar acciones a dicho objeto. Sería los *hilos* que manejan la marioneta. Para poder ceñirnos a esta filosofía debemos definir los nodos VRML de nuestras aplicaciones en base a un formato definido, no basta conocer la especificación VRML-EAI, debemos seguir las normas fijadas para la integración con XMMVR.

5. Conclusiones y trabajo futuro

Demostramos la necesidad de definir un meta-guión para especificar cualquier mundo virtual que permita interacción multimodal. Aportando modularidad, reutilización y estandarización. Para comprobar la efectividad del lenguaje propuesto, hemos implementado la arquitectura descrita con una pequeña aplicación de ejemplo, en la que sólo hemos definido un actor y un escenario. Habría que aumentar el número de actores y escenarios en futuros desarrollos. Sólo consideramos la metáfora del proxy o delegado para la interacción vocal, debemos dar solución a cada una de las metáforas de interacción vocal o a todas globalmente. Hemos utilizado VRML para especificar los elementos y la interacción gráfica basándonos en un navegador ya obsoleto. Redefiniremos nuestra arquitectura para poder trabajar con el lenguaje X3D utilizando un navegador adecuado que requerirá un desarrollo basado en el API SAI y se basará en software de código abierto. Finalmente evaluaremos las capacidades de interacción definidas con usuarios reales para corregir y mejorar nuestra propuesta.

Agradecimientos

Trabajo financiado parcialmente por la Consejería de Educación de la Junta de Castilla y León. Proyecto VA053A05 ARACNOS: MARCOS PARA EL DESARROLLO DE INTERFACES WEB 3D QUE INCORPOREN INTERACCION VISUAL Y HABLADA CON EL USUARIO.

Referencias

[1] W.R.Sherman, A.Craig. *Understanding Virtual Reality: Interface, Application, and Design*, Morgan Kaufmann, 2002
[2] D.Dahl. *Practical Spoken Dialog Systems (Text, Speech and Language Technology)*, Springer, 2004
[3] C.González y otros. *Incorporación de interacción vocal en mundos virtuales usando VoiceXML*, CEIG, 2004
[4] VoiceXML: http://www.voicexml.org
[5] X3D: http//www.web3d.org/x3d.html
[6] J.Hartman, J.Wernecke. *The VRML 2.0 Handbook*, Silicon Graphics, 1994
[7] R.Dachselt. *Action Spaces - A metaphorical concept to support navigation and interaction in 3D interfaces*; D.U.T., 2000
[8] S.McGlashan, T.Axling. *Talking to Agents in Virtual Worlds*, UK VR-SIG Conf., 1996
[9] XMMVR http://www.xmmvr.info
[10] ATLAS IBERVOX http://www.verbio.com