



Óscar Belmonte Fernández<sup>1</sup>, Miguel Castañeda Garay<sup>2</sup>, David Fernández Mota<sup>1</sup>, José Gil Altava<sup>1</sup>, Sergio Aguado Gonzalez<sup>1</sup>, Ester Varella Cerdá<sup>1</sup>, Manuela Nuñez Redó<sup>1</sup>, José Vicente Segarra Sola<sup>1</sup>

<sup>1</sup>Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I de Castellón; <sup>2</sup>Departamento de Computación, Universidad de Santiago de Cuba

<oscar.belmonte@uji.es>

## 1. Introducción

El objetivo de la simulación es obtener información de un sistema a partir de un modelo que represente el sistema y no directamente a partir del sistema. La simulación de sistemas es útil en casos donde no es posible experimentar directamente con el sistema real, la experimentación es peligrosa, demasiado cara, o no hay un sistema real sobre el que experimentar. La simulación de sistemas se utiliza con éxito en campos tales como la Ingeniería, la Arquitectura, Medicina, Física, Educación y en los negocios. Una simulación puede tener más de un componente simulado, cada uno de estos componentes coopera con el resto para completar la simulación. Un caso particular de simulación con más de un componente es un sistema distribuido, donde los componentes participantes en la simulación están geográficamente distribuidos. Estos sistemas software son llamados Entornos Virtuales Distribuidos (DVE de las siglas en inglés) [1]. Dos aspectos importantes en un DVE son, mantener la consistencia entre los componentes que toman parte en la simulación y al mismo tiempo minimizar la información intercambiada entre estos componentes. La simulación debe ser también escalable y robusta cuando se añaden nuevos componentes. En este último caso, es importante

# Un sistema de Realidad Virtual distribuido basado en la gestión de recursos federados

**Resumen:** la gran mayoría de los entornos virtuales distribuidos utilizan protocolos ad-hoc para compartir recursos y sincronizar la interacción de varios usuarios sobre un mundo virtual común. En este escenario, es difícil incorporar nuevos dispositivos de interacción si desde el principio del diseño del entorno virtual no se han tenido en cuenta. En este artículo se muestra cómo gestionar dispositivos de Realidad Virtual como recursos federados en un mundo virtual utilizando la arquitectura estándar HLA-RTI. Esta aproximación se ha utilizado como marco de trabajo para construir simuladores para el entrenamiento de operarios en ingeniería civil.

**Palabras clave:** dispositivos de interacción virtuales, entornos interactivos, gráficos distribuidos, Realidad Virtual.

tomar en consideración aspectos tales como la arquitectura de las comunicaciones, la suscripción a un grupo de interés, el control de la concurrencia, la replicación de los datos en los componentes y el balanceo de carga [2].

La visualización realista de entornos virtuales ha alcanzado un alto grado de desarrollo gracias a la rápida evolución del hardware gráfico y a las excelentes bibliotecas gráficas disponibles en la actualidad, como por ejemplo OpenGL y Direct3D. Sin embargo, el modo de programar estas bibliotecas gráficas está muy cercano al proceso aplicado a los datos en la tubería gráfica. Estas bibliotecas no proporcionan funcionalidades de alto nivel para el desarrollo de aplicaciones gráficas, tales como grafos de escena, detección de colisiones o simulaciones físicas. No existe un estándar de facto que proporcione tal funcionalidad, aunque existen un gran número de APIs que intentan cubrir esta carencia. Ha aparecido un elevado número de soluciones que cubren estas carencias, y éstas pueden

ser clasificadas en dos grandes grupos:

- Grafos de escena.
- Motores gráficos para el desarrollo de vídeo-juegos.

Un grafo de escena utiliza una estructura en árbol para describir una escena. Los nodos del árbol describen elementos visuales u operaciones sobre estos elementos. Operaciones típicas son las transformaciones geométricas, nodos de selección, nivel de detalle, etc. Un ejemplo de este tipo de soluciones es OpenSceneGraph, una implementación en C++ de grafo de escena de código abierto. Otro ejemplo es Java3D, grafo de escena desarrollado en Java. En los motores de juegos además de la visualización como en el caso de los grafos de escena, también se proporcionan motores físicos con los que se obtiene un comportamiento de los objetos en la escena altamente realista, como la detección de colisiones, animación y sonido 3D. La integración de estas características hace a los motores de juegos candidatos interesantes en

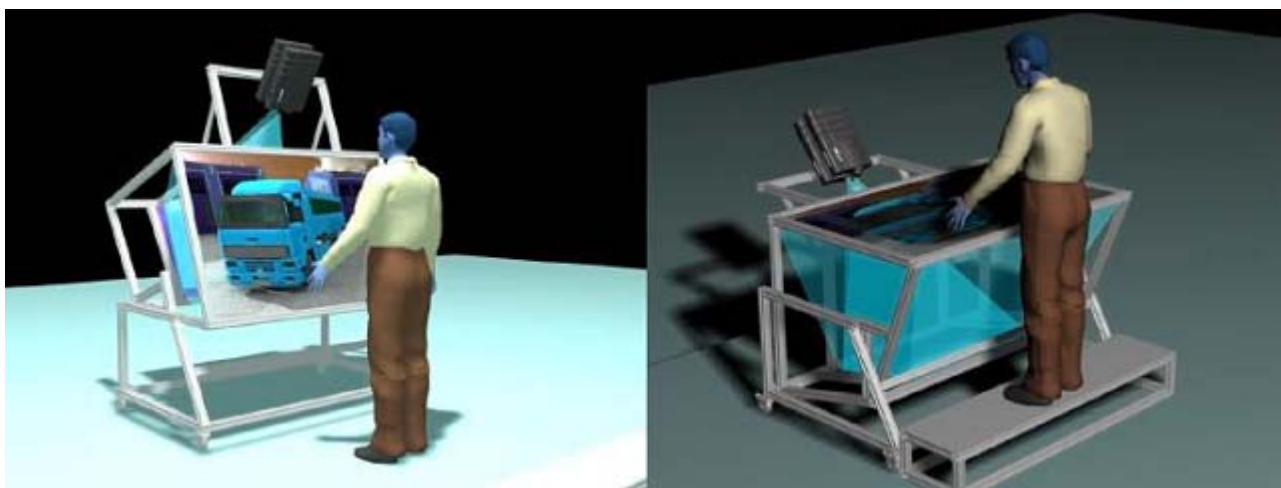


Figura 1. El sistema de RV.

los que basar el desarrollo de aplicaciones de Realidad Virtual (RV). Algunos de los motores de juegos más conocidos son: Torque, jMonkey Engine, Ogre, y Doom Engine.

Este trabajo presenta el resultado de un proyecto de investigación y transferencia de tecnología desarrollado en la Universitat Jaume I. El objetivo de este proyecto es construir un prototipo de simulador basado en RV para ser utilizado en el entrenamiento de trabajadores especializados, especialmente en ingeniería civil de obra subterránea. No obstante, el uso del simulador es genérico y puede adaptarse a la simulación de cualquier tipo de entornos virtuales. El sistema de RV cuenta con el siguiente equipamiento:

- Un sistema de proyección estereoscópica. Un sistema articulado permite orientar la pantalla de proyección vertical u horizontalmente. La **figura 1** muestra este sistema.
- Un sistema de posicionamiento basado en cámaras infrarrojas. El sistema consiste en cinco cámaras infrarrojas colocadas en las cuatro esquinas de la pantalla de proyección y una quinta cámara colocada en posición azimutal con respecto al observador. Este sistema permite conocer la posición del observador y la dirección de su mirada.
- Un sistema de interacción basado en guantes de datos que permiten interaccionar con los objetos de la escena.
- Un sistema de interacción *multi-touch* basado en visión artificial. Este sistema utiliza una cámara infrarroja para detectar la acción del usuario directamente sobre la pantalla de proyección.
- Un sistema de navegación virtual basada en dispositivos Sunspot.
- Un sistema de sonido envolvente 5.1. Con este sistema el usuario recibe información y realimentación sonora.
- Un sistema de información al usuario *text-to-speech*. A través de este sistema el usuario recibe instrucciones e información en lenguaje natural.

## 2. Descripción del sistema

### 2.1. Requerimientos del sistema

#### 2.1.1. Dispositivos de RV heterogéneos

Existe una gran cantidad de dispositivos de RV que pueden ser utilizados en una simulación. Un objetivo de nuestro sistema es que cualquier dispositivo pueda formar parte en la simulación como un componente. El sistema no está restringido a un grupo de dispositivos, si no que su diseño permite integrar cualquier dispositivo en la simulación. Se puede decir que el sistema de simulación es independiente de los dispositivos utilizados. La arquitectura utilizada para conseguir este objetivo crea una capa de abstracción entre el protocolo de comunicación y los dispositivos de RV, de tal modo que cualquier dispositivo que implemente un determinado interfaz puede formar parte de la simulación.

#### 2.1.2. Distribuido

Es común que los componentes que forman parte de una simulación se encuentren geográficamente dispersos. Estos componentes sincronizan su estado durante la ejecución de la simulación a través de líneas de comunicación de datos. Un requerimiento del sistema que se ha desarrollado era que fuese capaz de realizar simulaciones en un entorno distribuido. En este caso es también importante considerar la escalabilidad del sistema [2].

#### 2.1.3. Flexibilidad para implementar nuevos escenarios

Otro requerimiento importante era que el sistema de simulación no estuviese enfocado en un tipo de simulación en concreto, por el contrario, el sistema debía ser lo suficientemente flexible y abierto para permitir un amplio espectro de modelos simulados en el campo de entrenamiento de personal especializado. El sistema fue diseñado para ser capaz de realizar simulaciones para robots guiados, paseos virtuales, conducción de vehículos como camiones, tractores, excavadoras, etc.

## 2.2. Sistema de simulación

El paso de los sistemas de simulación tradicionales con un único componente a simulaciones distribuidas, abiertas e interactivas con múltiples componentes requiere el desarrollo de nuevas técnicas. En 1995 HLA fue propuesto por la Oficina de Modelado de la Defensa y Simulación del Departamento de Defensa de Estados Unidos, con el objetivo de proporcionar un enlace que permitiese a múltiples simulaciones individuales comunicarse entre sí con el objetivo de construir una simulación global. Finalmente, HLA se convirtió en un estándar de simulación (IEEE 1516 [3]), de modo que este estándar se puede utilizar en otros campos además del estrictamente militar, como la educación, el entretenimiento o la ingeniería.

## 2.3. Descripción del sistema de simulación

### 2.3.1. Escenario de RV

La Realidad Virtual se define como un medio compuesto por una simulación de ordenador altamente interactiva, que registra la posición del observador y sustituye o aumenta la realimentación de uno o más sentidos para dar la sensación de estar inmerso o presente en la simulación [4]. En este proyecto se utiliza un sistema de RV semi-inmersivo. El sistema de visualización está compuesto por una pantalla de 169 x 98 cm. sobre la que se proyectan dos imágenes con el objeto de obtener visualización estereoscópica. Cada una de las dos imágenes es generada por un proyector de alta resolución. Se han modelado cuatro escenarios para evaluar el objetivo propuesto. El primer escenario está compuesto por una tuneladora; utilizando los controles el usuario puede mover la cabeza de la tuneladora e incrementar o reducir su velocidad. El segundo y tercer escenario son simulaciones de paseos virtuales; el primero es un paseo virtual por el campus de la Universitat Jaume I y el segundo es un paseo por un túnel. El último escenario simula intervenciones de un robot submarino dotado con un brazo robótico de cuatro grados de libertad.

Para la generación de los gráficos se utilizó el motor de juegos *jMonkey Engine* (jME). JME es un API basado en un grafo de escena de alto rendimiento. Todos los nodos son genéricos incluyendo tanto información geométrica y transformaciones. Los nodos hoja del grafo de escena representan geometría, jME soporta parches de Bezier, líneas, puntos, modelos 3D, terrenos y nivel de detalle entre otros.

Con el objeto de poder utilizar HLA en el sistema de RV se han debido realizar algunas modificaciones a la implementación de jME. jME proporciona una implementación básica (*Simple*) basada en las siguientes capas de abstracción:

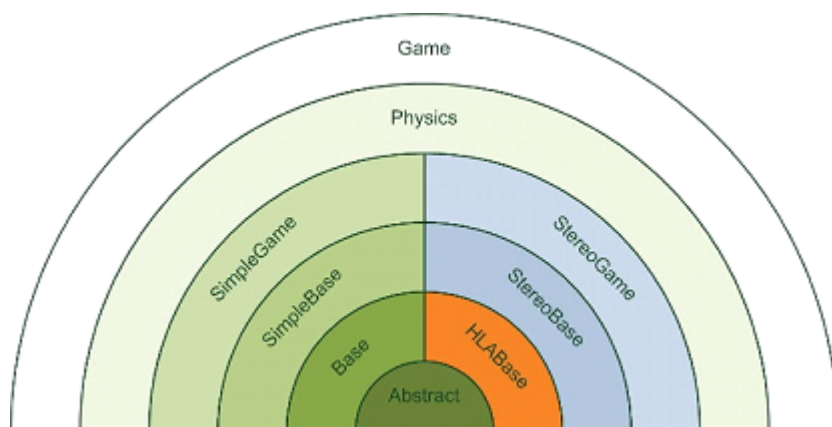


Figura 2. Las capas HLA introducidas en jMonkey Engine.

- **Abstract layer:** Definición básica y sistema de configuración.
- **Base layer:** Implementación del bucle principal.
- **SimpleBase layer:** Definición de los elementos básicos del grafo de escena.
- **SimpleGame layer:** Definición de la secuencia de actualización por eventos y de la visualización de la escena.
- **Physics layer:** Definición e inicio del motor de física.

Esta implementación se modificó para incluir en el motor de juegos tanto la estereoscopia como el protocolo para realizar simulaciones distribuidas. Para incluir estereoscopia se modificaron las capas SimpleBase y SimpleGame de modo que además de incluir estereoscopia se mantuviese la compatibilidad entre el modelo original (*Simple*) y las capas superiores. La **figura 2** muestra las modificaciones realizadas sobre la estructura original:

- **StreoBase:** Esta capa añade los elementos básicos para obtener estereoscopia y proporciona simplicidad a los elementos que pueden ofrecer una mejor definición en las capas superiores.
- **StereoGame:** Implementa la visualización estereoscópica.

Siguiendo la misma metodología se implementó una nueva capa HLABase la cual permite que los componentes que toman parte en la simulación estén sincronizados estableciendo la base para una simulación distribuida.

El bucle principal del juego está compuesto por varias etapas: actualización de datos de entrada, actualización del estado, procesado de la escena y visualización. Con la implementación de HLABase, todos los federados necesitan conocer cada etapa del

bucle de juego antes de que puedan avanzar sus estados y, como resultado, todo el sistema resulta sincronizado de modo transparente durante la ejecución. Sin embargo, la velocidad de avance de la simulación será la velocidad de avance del componente más lento de la federación. De este modo, HLA redefine la implementación del bucle de juego con las siguientes etapas:

- Inicialización del sistema mediante HLA: registro de federados objetos compartidos.
- Control de los objetos compartidos y envío de las actualizaciones: si un nuevo cambio se produce, el sistema toma el control del objeto y envía sus actualizaciones al resto de la federación.
- Sincronización de todos los federados y recepción de los datos: se establece un nuevo punto de sincronización para hacer avanzar la federación de tal modo que se asegure que todos los federados han enviado sus actualizaciones a la federación antes de seguir con el bucle del juego.
- Actualización de los objetos compartidos.
- Finalización del sistema HLA.

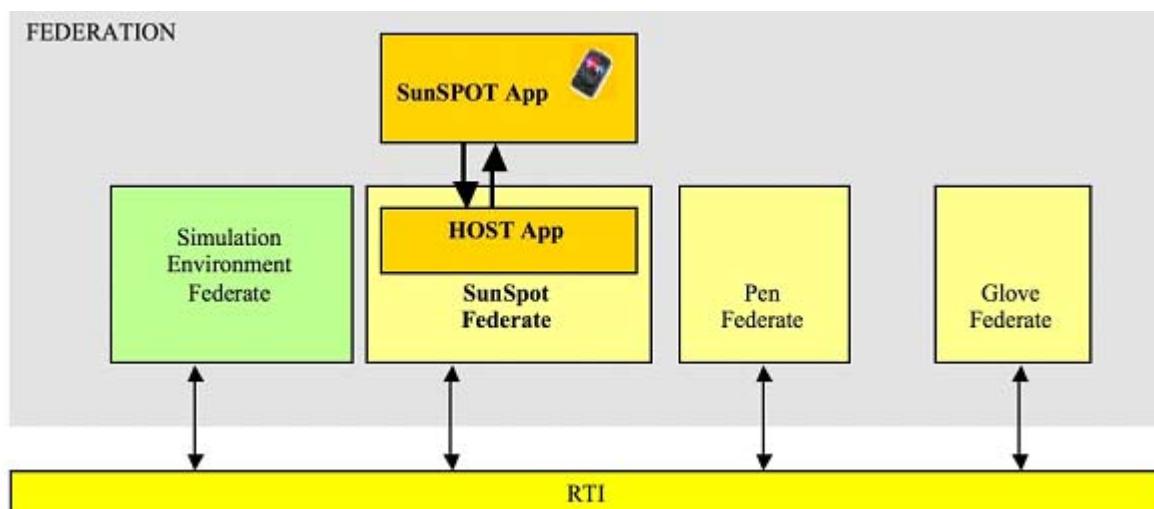
Durante una simulación, resulta muy interesante que el usuario obtenga una realimentación continua por parte del sistema en forma de mensajes. Para ello se ha diseñado un módulo capaz de realizar locución de mensajes al tiempo que estos se muestran por pantalla. Los mensajes de audio se generan utilizando un sistema de Texto a Voz. Existen dos métodos para mostrar información textual en pantalla:

- La información textual es un objeto más de la escena; de este modo, se puede añadir información textual a un objeto colocando uno de estos mensajes cerca de él.
- **Heads-Up Display:** el objetivo de este método es mostrar al usuario información relevante para que este actúe en consecuencia.

La estereoscopia es la técnica que permite crear en el usuario la ilusión de que está viendo una escena tridimensional al presentarle imágenes distintas para cada uno de sus ojos. La estereoscopia puede ser pasiva o activa. En el caso de la estereoscopia pasiva se utilizan un par de proyectores delante de los cuales se colocan filtros polarizadores ortogonales, y el usuario viste unas gafas con un filtro polarizador para cada uno de los ojos de tal modo que finalmente cada ojo sólo verá la imagen proveniente de uno de los proyectores, la otra quedará bloqueada por el filtro. En el caso de la estereoscopia activa, se utiliza un solo proyector con una frecuencia de refresco lo suficientemente elevada como para generar imágenes alternativas para el ojo izquierdo y el ojo derecho. Por otro lado, el usuario viste unas gafas con cristales que pueden oscurecerse de manera sincronizada con el proyector, de tal modo que cuando el proyector genera la imagen para el ojo derecho genera a su vez una señal de sincronía que oscurece el cristal delante del ojo izquierdo para que la imagen no llegue a él y viceversa.

En el simulador desarrollado se ha utilizado la técnica de estereoscopia pasiva. Para obtener las dos imágenes necesarias para la estereoscopia se utilizó el escritorio extendido con un único lienzo con tamaño de ancho el doble del tamaño de la imagen, a la izquierda del lienzo se dibuja la imagen correspondiente al ojo izquierdo y a la derecha la imagen para el ojo derecho.

Para ofrecer una mejor sensación de inmersión en el escenario virtual se utilizó un sistema de sonido 5.1 donde cada canal se coloca alrededor del usuario y cada uno de ellos emite un sonido distinto, dependiendo de donde provenga el sonido original con respecto al observador. El sistema es programado mediante OpenAL una biblioteca de genera-



**Figura 3.** Integración de los dispositivos de RV utilizando HLA-RTI.



**Figura 4.** Posición de los sensores en los guantes de datos.

ción de sonido posicional y multicanal que se integra perfectamente con jMonkey Engine.

### 2.3.2. Integración de los dispositivos de RV.

La **figura 3** muestra cómo los dispositivos de RV se definen como federados utilizando HLA-RTI.

Los guantes de datos usados para ambas manos miden la curvatura de las articulaciones de los dedos. Cada guante de datos tiene 14 sensores en las posiciones de la parte inferior mostradas en color (verde) en la **figura 4**, las 4 posiciones superiores mostradas en color (azul) no representan sensores reales, sus valores son los mismos que los valores de los sensores en el correspondiente dedo, se han introducido para mejorar la representación gráfica de la mano virtual. Cada guante de datos se puede conectar al ordenador a través de un puerto serie o bien a través una conexión *bluetooth*.

El modelo 3D representa una mano con las mismas articulaciones que una mano real. La mano virtual se ha modelado con 19 huesos recubiertos por piel. La malla que representa la piel se modeló de tal modo que su movimiento está sincronizado con el de los huesos a los que recubre. jMonkey Engine permite mover cada articulación de modo separado con los valores proporcionados por el guante de datos.

Los guantes de datos proporcionan un API programado en C++ al que no es posible acceder directamente desde Java. Para obtener acceso al API en C++ se desarrolló una biblioteca en Java que conecta la biblioteca nativa en C++ y proporciona la misma funcionalidad. La comunicación entre Java y C++ se consiguió a través de JNI (*Java Native Interface*).

El objetivo del seguimiento de la dirección de visión es actualizar el punto de vista de la escena a partir de los movimientos de la cabeza del usuario. El movimiento de la cabeza es capturado por un conjunto de cinco cámaras infrarrojas que capturan el movimiento de marcadores infrarrojos colocados

sobre una gorra que viste el usuario. A partir de la información capturada es posible determinar la posición del observador y su dirección de mirada. Esta información es convertida en movimientos de la cámara dentro del escenario virtual.

La comunicación con las cámaras infrarrojas se consigue de un modo similar al utilizado con los guantes de datos. El API que ofrece acceso a las cámaras está escrito en C++, por lo que es necesario una biblioteca Java que utilice JNI para comunicarse con ella.

Para obtener la posición del guante en el espacio 3D se utiliza el mismo sistema de seguimiento que el utilizado para capturar el movimiento de la cabeza. Sobre el guante se colocan marcadores infrarrojos a partir de los cuales es posible obtener la posición y orientación del guante en el espacio 3D, y el usuario puede interactuar directamente con los objetos del escenario virtual.

Los dispositivos SunSpot desarrollados por Sun Microsystems son dispositivos inalámbricos que poseen una serie de sensores, en particular un acelerómetro capaz de medir la inclinación del dispositivo con respecto de los ejes coordenados. Esta información es utilizada para proporcionar al usuario la capacidad de moverse en el escenario virtual inclinando el dispositivo hacia adelante para avanzar, hacia atrás para retroceder, o hacia la izquierda y derecha para efectuar cambios en la dirección de avance. Esta información es transmitida mediante el protocolo ZigBee.

El dispositivo IR Pen permite al usuario la interacción directa con la pantalla de proyección a través de un puntero infrarrojo. El puntero está construido con un LED infrarrojo que se ilumina con la pulsación de un botón. Una cámara infrarroja detecta la posición del LED sobre la pantalla y esta información es procesada para convertirla en coordenadas X, Y sobre la pantalla. La aplicación es capaz de detectar la luz infrarroja proveniente de múltiples LEDs lo que permite convertir a la pantalla de proyección en una pantalla *multi-touch*.

Las pantallas de proyección de gran formato son ampliamente utilizadas en aplicaciones de RV [5]. En muchos casos, estas pantallas no funcionan únicamente como dispositivos de salida, sino que también funcionan como dispositivos de entrada a través de la incorporación de la detección del tacto. Las tecnologías utilizadas en las pantallas de proyección de gran formato táctiles son: medida de la capacitancia, medida de la resistencia, tecnología ultrasónica y tecnología infrarroja entre las principales. Sin embargo, la tecnología que recientemente está atrayendo gran interés es la tecnología basada en visión por computador [6]. Utilizando esta tecnología, la

interacción del usuario es detectado por un sistema de cámaras, y los gestos son interpretados como acciones. Otra característica interesante es que la resolución no depende del tamaño de la pantalla, sino de los proyectores utilizados. El hecho de que múltiples usuarios puedan interactuar al mismo tiempo es la característica que las hace más atractivas en aplicaciones de RV.

El sistema de RV que se ha desarrollado está equipado con una cámara infrarroja capaz de capturar perturbaciones fotométricas al tocar la pantalla, y estas perturbaciones son interpretadas como gestos del usuario.

### 2.3.3. Simulación distribuida

En esta sección se presentan las modificaciones realizadas en el bucle de juego para permitir la sincronización de todas las acciones provenientes de los distintos dispositivos de RV utilizando el protocolo HLA. Para evitar la restricción de velocidad de actualización debida al miembro más lento de la federación se decidió desarrollar un sistema bajo el protocolo HLA centrado en los dispositivos y con un alto nivel de abstracción. Cada dispositivo proporciona sus datos en un sistema secuencial de cuatro estados que interactúa directamente con la arquitectura para enviar los datos.

Este sistema de control, integrado en la clase Federate, puede estar en alguno de los siguientes estados:

- Estado de control: El dispositivo tiene un nuevo dato que enviar, negocia el control del objeto compartido.
- Estado de envío: Si se ha obtenido el control, el dispositivo envía el dato, si no, mantiene el dato hasta el siguiente estado de control.
- Estado de recepción: El dispositivo espera hasta que todos los federados han recibido las actualizaciones. Este es el punto de sincronización entre los federados.
- Estado de actualización: El dato es enviado a la aplicación conectada.

La sincronización no afecta al bucle principal de juego y elimina la restricción en la velocidad de actualización. La restricción queda limitada a la actualización de los dispositivos, donde su velocidad se ajustará a la velocidad de las comunicaciones a través del protocolo HLA que procesa los datos. La **figura 5** muestra como funciona el control del sistema.

Aunque cada dispositivo define sus propios datos específicos, un tipo de dato genérico (SharedData) aparece en las comunicaciones. Las comunicaciones pueden ser simplificadas utilizando un tipo de datos genérico teniendo en cuenta que son traducidas con anterioridad.

Para cada nuevo dispositivo que se desee

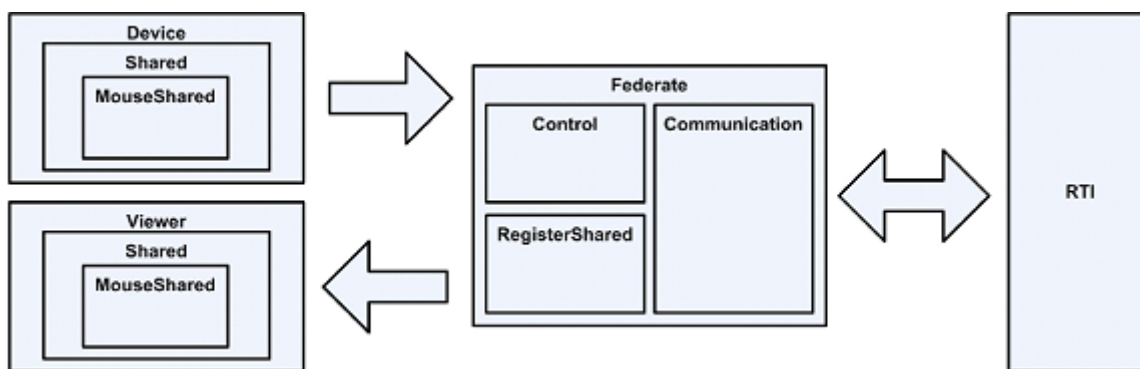


Figura 5. Simulación mediante HLA.

conectar al sistema es necesario realizar los siguientes tres pasos:

- Definir los atributos de comunicación del dispositivo.
- Implementar el tipo de dato específico.
- Implementar un método de codificación y decodificación que traduzca el dato entre el tipo genérico y el tipo específico.

De este modo, la recepción, el envío y la gestión de los datos se hará sobre el tipo de datos genérico (SharedData).

Las clases del proyecto son:

- Shared: Tipo genérico parametrizado, permite la conversión al tipo de datos común (SharedData) por parte de cada uno de los dispositivos de RV para ser enviado sobre el protocolo RTI.
- RegisterShared: Registra el dispositivo conectado a cada federado y gestiona la recepción y el envío de eventos.
- Federate: Controla y dirige la comunicación con RTI.

La implementación del protocolo HLA utilizada es Portico [7]. Portico ofrece un entorno extensible para simulaciones basadas en HLA orientadas al desarrollo y la investigación.

Para describir cada uno de los dispositivos en el entorno de simulación bajo el protocolo HLA-RTI son necesarios los siguientes pasos:

- Definir los atributos del objeto específicos que interactuarán en el sistema de RV.
- Implementar una estructura que gestione la información.
- Implementar un método de codificación y decodificación que realice la traducción entre el tipo de datos específico y el tipo de datos genérico.

Únicamente existe una descripción de la federación utilizada para agrupar a todos los federados, no obstante, la información en la descripción se puede restringir utilizando regiones de interés o suscripciones para cada uno de los federados.

### 2.3.4. Aplicaciones de ejemplo

Se han implementado cuatro entornos virtuales distribuidos. La **tabla 1** muestra los dispositivos de RV utilizados en cada uno de estos cuatro entornos.

En el caso del simulador de robot submarino, el modelo geométrico está formado por un brazo y una mano articulada montados sobre un robot submarino. El objetivo de este simulador es el entrenamiento de personal especializado, y la planificación de intervenciones submarinas.

Todos los dispositivos de control del robot en la simulación son virtuales. El operario que controla el robot viste un par de guantes de datos con marcadores infrarrojos. En la parte

superior izquierda de la **figura 6** se muestra este dispositivo. Los guantes de datos leen los gestos de la mano del operario, y la lectura de los marcadores por parte de las cámaras infrarrojas devuelven la posición y orientación de las manos en el espacio 3D. La secuencia de movimientos necesaria para manipular un *joystick* virtual con el que controlar el robot es totalmente virtual. Con el *joystick* virtual de la mano izquierda se controla el movimiento del robot, el *joystick* virtual de la mano derecha sirve para controlar el brazo y la mano robóticos.

### 3. Conclusiones y trabajo futuro

En el momento en el que se desarrolló este trabajo la implementación Portico de HLA era incompleta, las mejoras que está planeado que se introduzcan en esta implementación son:

- Gestionar la toma de posesión de los atributos.
- Mejorar la escalabilidad del sistema, de tal modo que el servidor RTI se ejecute en más de un ordenador.

Esperamos que con la introducción de estas nuevas mejoras nuestros simuladores alcancen un mejor rendimiento y se extiendan sus posibilidades de uso.

### Agradecimientos

Este proyecto de investigación ha sido financiado por la Fundació Caixa Castelló – Bancaixa y la

Entorno Virtual Distribuido.	Dispositivos de RV utilizados.
Prototipo de tuneladora.	IR Pen, Posicionamiento, SunSpot, Sonido 3D.
Paseo virtual por la Universidad Jaume I.	IR Pen, Posicionamiento, SunSpot, Sonido 3D.
Paseo virtual por un túnel en construcción.	IR Pen, Posicionamiento, SunSpot, Sonido 3D.
Simulador de robot submarino.	IR Pen, Posicionamiento, SunSpot, Sonido 3D, Guantes de datos.

Tabla 1. Dispositivos de RV utilizados en cada uno de los entornos.



Figura 6. Modelo del robot submarino.

Universitat Jaume I ((P1-1B2009-50), y por el Ministerio de Ciencia e Innovación español (PSS-380000-2008-21, La Ciudad Multidimensional).

Queremos dar las gracias a Gianluca de Novi por proporcionarnos el modelo del robot submarino; al grupo de investigación EGICAD de la Universidad de Cantabria por proporcionarnos el modelo de la tuneladora, y al grupo de investigación GITECO de la Universidad de Cantabria por proporcionarnos el modelo del túnel.

### Referencias

- [1] S. Singhal, M. Zyda. *Networked virtual environments? Design and Implementation*. Addison-Wesley, New York, 1999. ISBN-10: 0201325578.
- [2] Dongman Lee, Mingyu Lim, Seunghyun Han, Kyungmin Lee. ATLAS: A scalable network framework for distributed virtual environments. *Presence* 16 (2), 2007, pp: 125-156.
- [3] IEEE. IEEE standard for modeling and simulation (M&S): High level architecture (HLA), <<http://standards.ieee.org>>
- [4] William R. Sherman, Alan B. Craig. *Understanding Virtual Reality: Interface, Application, and Design*. Morgan Kaufmann, San Francisco, EUA, 2003.
- [5] T. Ni, G. Schmidt, O. Staadt, M.A. Livingston, R. Ball, R. May. A survey of large high-resolution display technologies, techniques, and applications. *Proceedings of the IEEE Virtual Reality Conference, VR'06*, 2006.
- [6] D. Cavens, F. Vogt, S. Fels, M. Meitner. Interacting with the big screen: Pointers to ponder. *Proceedings of ACM conference on Computer Human Interaction, CHI 2002*, Minneapolis, Minnesota, USA, 2002, pp. 678-679.
- [7] Portico project. Disponible en <[http://www.porticoproject.org/index.php?title=Main\\_Page](http://www.porticoproject.org/index.php?title=Main_Page)>.



## LISTAS DE DISTRIBUCIÓN DE ATI

¿Tienes dudas sobre cuestiones técnicas relacionadas con este u otros temas?

¿Quieres comentar con otros colegas aspectos generales de la profesión?

En nuestras listas de distribución tenemos datos de alta cerca de 500 profesionales, todos ellos socios de ATI, con quienes contactar y comunicarte.

Cuestiones generales sobre la profesión informática y otros temas >> [foro@ati.es](mailto:foro@ati.es)

Comentarios, dudas o sugerencias técnicas >> [tecnica@ati.es](mailto:tecnica@ati.es)

\* Para darse de alta en estas listas hay que ser socio de ATI y acceder a la opción Listas de distribución del menú de la intranet de ATI.

\*\* Para el acceso a la intranet contactar con [secregen@ati.es](mailto:secregen@ati.es).

ATI Secretaría General | Via Laietana 46, ppal. 1a. | 08003 Barcelona | 93 412 52 35 | [secregen@ati.es](mailto:secregen@ati.es) | [www.ati.es](http://www.ati.es)