

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DEPARTAMENTO DE COMUNICACIONES



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

TESIS DOCTORAL

“ARQUITECTURA DE INTERCONEXIÓN ENTRE SISTEMAS  
C4ISR Y MUNDOS VIRTUALES APLICANDO EL  
ESTÁNDAR MPEG-V”

**Autora:** Laura Inés Ardila Sierra

**Director:** Dr. Manuel Esteve Domingo

VALENCIA, OCTUBRE DE 2015



# Agradecimientos

Los últimos años han sido probablemente los más desafiantes de mi vida, a la vez que los más enriquecedores. He tenido que levantarme tras muchos momentos difíciles, y me siento inmensamente agradecida con todos quienes me han dado motivos para sonreír y fuerzas para seguir adelante.

Quiero agradecer al Dr. D. Manuel Esteve por hacer posible que hoy esté finalizando esta tesis doctoral. Gracias, Manuel, por tu orientación, apoyo y paciencia. Han sido muchos los obstáculos, pero lo hemos conseguido.

Gracias también al Dr. D. Carlos Palau, quien, sin tener ninguna obligación, está siempre dispuesto a echar una mano. Al Dr. D. Israel Pérez por sus aportes a esta investigación, y a mis compañeros del Grupo de Sistemas y Aplicaciones de Tiempo Real Distribuidos por todos los buenos momentos.

Al Coro de la UPV por ser mi refugio, mi familia adoptiva, y por darme la oportunidad de participar en tantos maravillosos proyectos. A mis *compis* de cuerda; sois las mejores.

Finalmente a mis padres, hermanas, y demás familiares y amigos, de aquí y de allá, por su constante e incondicional apoyo. Siempre están conmigo, porque el cariño no entiende de distancias.

A todos quienes habéis contribuido a que hoy esté terminando esta etapa, muchísimas gracias.



# Resumen

Cuando ocurre una emergencia, el personal involucrado en su resolución hace todo lo posible por reducir al mínimo las fatalidades y mantener seguros a los ciudadanos. La realización de ejercicios de entrenamiento de forma continua y regular ayuda a mejorar la planificación, los protocolos de actuación, y a armonizar los procedimientos entre diferentes agencias.

La aplicación de las tecnologías de información y comunicaciones (TIC) en la gestión de grandes emergencias es un campo de investigación muy relevante. En particular, el desarrollo de sistemas de mando y control (C2IS) para gestionar la resolución de una crisis y el uso de mundos virtuales para la implementación de ejercicios de entrenamiento de personal son áreas en continuo desarrollo. Sin embargo, la integración de un C2IS dentro de un sistema de simulación en red para entrenamiento basado en mundos virtuales y usando equipos comerciales (COTS) no ha sido explotada suficientemente.

Dentro de este contexto, desarrollar un sistema híbrido conectando entidades y sensores reales con entidades virtuales presenta una estrategia clave. En particular, la implementación de un software intermedio (*middleware*) para interconectar aplicaciones heterogéneas usando formatos de datos estandarizados.

Esta tesis contribuye a facilitar la realización de ejercicios de entrenamiento en gestión de emergencias que en la vida real son complicados, poco flexibles y muy costosos.

Se propone, implementa y valida una arquitectura para conectar un sistema de mando y control operando en el mundo real con uno o varios mundos virtuales, a través de una pasarela de interconexión que implementa un modelo de datos basado en MPEG-V y utiliza servicios web vía TCP/IP para la comunicación entre nodos.

# Abstract

Whenever an emergency occurs, personnel involved in its resolution make every possible effort to minimize casualties and keep citizens safe. Performing continuous and regular training exercises helps improve planning, actuation protocols and the harmonization of procedures between various agencies.

The application of information and communication technologies (ICT) in emergency management is a relevant research field. Particularly, the development of command and control systems (C2IS) to manage crisis resolution and the use of virtual worlds to develop training exercises are challenging and evolving areas. However, the insertion of a C2IS within a networked training system based on virtual worlds and using commercial off-the-shelf equipment has not been sufficiently exploited.

In this context, the development of a hybrid system connecting real computers and sensors with virtual computers is a key strategy. In particular, the deployment of a middleware to interconnect heterogeneous applications using standardized data formats.

This thesis contributes to facilitate performing training exercises in emergency management that in real life are complicated, inflexible and very expensive.

An architecture to connect a command and control system operating in the real world with one or several virtual worlds

through an interconnection gateway that implements MPEG-V based data formats and by using web services via TCP/IP as a communication platform is proposed, deployed and validated.



# Resum

Quan ocorre una emergència, el personal involucrat en la resolució fa tots els possibles per reduir al mínim les fatalitats i mantenir segurs els ciutadans. La realització d'exercicis d'entrenament de forma contínua i regular ajuda a millorar la planificació, els protocols d'actuació, i a harmonitzar els procediments entre diferents agències.

L'aplicació de les tecnologies d'informació i comunicacions (TIC) en la gestió de grans emergències és un camp de recerca molt rellevant. En particular, el desenvolupament de sistemes de comandament i control (C2IS) per gestionar la resolució d'una crisi i l'ús de mons virtuals per a la implementació d'exercicis d'entrenament de personal són àrees en continu desenvolupament. No obstant això, la integració d'un C2IS dins d'un sistema de simulació en xarxa per a entrenament basat en mons virtuals i usant equips comercials (COTS) encara no ha sigut prou explotada.

Dins d'aquest context, desenvolupar un sistema híbrid connectant ordinadors i sensors reals amb entitats virtuals suposa una estratègia clau. En particular, la implementació d'un programari intermediari (*middleware*) per interconnectar aplicacions heterogènies usant formats de dades estandarditzats.

Aquesta tesi contribueix a facilitar la realització d'exercicis d'entrenament en gestió d'emergències que en la vida real són complicats, poc flexibles i molt costosos.

Es proposa, implementa i valida una arquitectura per a connectar un sistema de comandament i control que opera en el món real amb un o diversos mons virtuals, a través d'una passarel·la d'interconnexió que implementa un model de dades basat en MPEG-V i utilitza serveis web via TCP/IP per a la comunicació entre nodes.

# Contenido

<b>Capítulo 1 Introducción y objetivos.....</b>	<b>19</b>
1.1. Introducción .....	19
1.2. Objetivos .....	23
1.3. Estructura de la tesis.....	25
1.4. Bibliografía .....	26
<b>Capítulo 2 Estado del arte .....</b>	<b>27</b>
2.1. Introducción .....	27
2.2. Sistemas de mando y control .....	28
2.3. Mundos virtuales.....	36
2.3.1. Representación de entornos virtuales .....	44
2.3.2. Arquitecturas.....	46
2.3.3. Ayuda y soporte a navegación .....	48
2.3.4. Adición de vida a los entornos.....	48
2.3.5. Interoperabilidad .....	50
2.4. Estándar MPEG-V.....	50
2.4.1. Términos, definiciones y abreviaturas .....	52
2.4.2. Partes del estándar.....	55
2.4.3. Aplicaciones y ejemplos .....	70
2.5. Bibliografía .....	71

<b>Capítulo 3 Diseño de la arquitectura .....</b>	<b>81</b>
<b>3.1. Introducción .....</b>	<b>81</b>
<b>3.2. Consideraciones base.....</b>	<b>82</b>
3.2.1. Generación de mundos virtuales .....	89
<b>3.3. El entrenador táctico .....</b>	<b>92</b>
<b>3.4. La pasarela de interconexión .....</b>	<b>97</b>
<b>3.5. Mecanismo de comunicación.....</b>	<b>100</b>
<b>3.6. Bibliografía .....</b>	<b>102</b>
<b>Capítulo 4 Implementación .....</b>	<b>105</b>
<b>4.1. Introducción .....</b>	<b>105</b>
<b>4.2. Modelo de datos .....</b>	<b>105</b>
4.2.1. Modelo de datos MPEG-V genérico.....	106
4.2.2. Modelo de datos MPEG-V para SIMACOP.....	110
4.2.3. Modelo de datos MPEG-V para OLIVE.....	111
<b>4.3. El entrenador táctico .....</b>	<b>114</b>
4.3.1. Servidor Entrenador Táctico.....	115
4.3.2. Cliente Entrenador Táctico.....	120
4.3.3. Servidor de Vídeo Virtual .....	122
<b>4.4. Comunicación entre nodos .....</b>	<b>125</b>
<b>4.5. Bibliografía .....</b>	<b>128</b>
<b>Capítulo 5 Validación.....</b>	<b>131</b>

<b>5.1. Introducción .....</b>	<b>131</b>
<b>5.2. Validación del sistema.....</b>	<b>131</b>
5.2.1. Consumo de CPU Servidor Entrenador Táctico.....	132
5.2.2. Consumo de ancho de banda Servidor Entrenador Táctico ....	133
5.2.3. Consumo de CPU Cliente Entrenador Táctico .....	138
5.2.4. Consumo de ancho de banda Cliente Entrenador Táctico .....	140
5.2.5. Consumo de CPU Servidor de Vídeo Virtual .....	142
5.2.6. Consumo de ancho de banda Servidor de Vídeo Virtual .....	145
<b>Capítulo 6 Conclusiones y desarrollos futuros .....</b>	<b>147</b>
<b>6.1. Conclusiones.....</b>	<b>147</b>
<b>6.2. Desarrollos futuros .....</b>	<b>150</b>
<b>Glosario de términos .....</b>	<b>153</b>
<b>Anexos.....</b>	<b>159</b>

# Lista de figuras

Figura 1. Modelo de un sistema de mando y control. ....	28
Figura 2. Estructura típica de un ICS. ....	30
Figura 3. Arquitectura del sistema C4ISR SIMACOP. ....	35
Figura 4. Continuo realidad – virtualidad en paralelo con el continuo grado de conocimiento del mundo. ....	39
Figura 5. Ejemplo de espacio combinado de realidad mixta. ....	40
Figura 6. Ejemplo de recorrido de un mundo virtual al mundo real. ...	41
Figura 7. Entrenamiento de first responders en OLIVE. ....	43
Figura 8. Ejemplo de arquitectura modular para mundos virtuales. ...	47
Figura 9. Arquitectura MPEG–V. ....	56
Figura 10. Sistema de coordenadas de referencia. ....	61
Figura 11. Mapeo de intenciones de mando. ....	62
Figura 12. Modelo de tiempo para metadatos de efectos sensoriales. ....	63
Figura 13. Alcance de MPEG-V parte 5. Caso 1: adaptación virtual a real. ....	66
Figura 14. Alcance de MPEG-V parte 5. Caso 2: adaptación real a virtual. ....	67
Figura 15. Arquitectura de un C4ISR. ....	86
Figura 16. Arquitectura MV con servidores asociados. ....	87
Figura 17. Mapeo semántico a alto nivel. ....	88

Figura 18. Principales componentes de la arquitectura de OLIVE. ...	90
Figura 19. Componentes principales del sistema. ....	92
Figura 20. Arquitectura de vídeo. ....	93
Figura 21. Arquitectura del sistema.....	94
Figura 22. Arquitectura del sistema ortogonal a la réplica del MR. ...	95
Figura 23. Ruta de los datos entre nodos. ....	96
Figura 24. Arquitectura en dos capas para la implementación de MPEG-V. ....	97
Figura 25. Ubicación de la pasarela de interconexión MR – MV. ....	98
Figura 26. Esquema conceptual de la pasarela de interconexión. ....	99
Figura 27. Arquitectura modular para la interconexión de sistemas.	100
Figura 28. Arquitectura de la pasarela de interconexión. ....	101
Figura 29. Mecanismo de comunicación.....	102
Figura 30. Definición del formato de datos genérico.....	107
Figura 31. Estructura elemento Alarm (alarma). ....	108
Figura 32. Estructura elemento Threat (amenaza). ....	108
Figura 33. Estructura elemento Position (posición). ....	109
Figura 34. Estructura elemento Sensor (sensor). ....	109
Figura 35. Estructura elemento Message (mensaje). ....	109
Figura 36. Definición formato de datos equivalente en MPEG-V para SIMACOP.....	110
Figura 37. Definición formato de datos equivalente en MPEG-V para OLIVE.....	111

Figura 38. Estructura elemento Avatar (avatar).....	112
Figura 39. Estructura elemento Product (producto).....	112
Figura 40. Estructura elemento Inventory (inventario).....	113
Figura 41. Estructura elemento VWO (objeto de mundo virtual). ....	113
Figura 42. Definición elemento Object (objeto) para OLIVE.....	114
Figura 43. Máquina de estados SET.....	119
Figura 44. CET. Pantalla principal.....	121
Figura 45. Transferencia de objetos del MR al MV.....	122
Figura 46. Vídeo del MV en el MR .....	123
Figura 47. SVV. Pantalla principal con 4 displays.....	124
Figura 48. Envío de datos con JSON.....	126
Figura 49. Ejemplo funcionamiento servicio web.....	128
Figura 50. Estructura de red de la plataforma de pruebas. ....	132
Figura 51. SET. Consumo de CPU. ....	134
Figura 52. SET. Uso ancho de banda pasarela MR-MV. ....	136
Figura 53. SET. Ancho de banda consumido en intercambio entre mundos virtuales. ....	137
Figura 54. SET. Comparación consumo estimado ancho de banda pasarela MR-MV vs. MV en función del número de clientes. ....	138
Figura 55. CET. Consumo de CPU para diferentes estados. ....	139
Figura 56. CET. Consumo ancho de banda.....	141
Figura 57. CET. Consumo de ancho de banda para la gestión de MV. ....	143



Figura 58. SVV. Consumo de CPU para 4 flujos de vídeo. ....	144
Figura 59. VVS. Estimación de evolución consumo de CPU para transmisión de vídeo con y sin grabación en función del número de flujos. ....	145
Figura 60. SVV. Consumo de ancho de banda.....	146

## Lista de tablas

Tabla 1. Configuración del servidor web y clientes.....	127
Tabla 2. Equipo utilizado en la plataforma de pruebas.....	133
Tabla 3. Consumo de CPU en el SVV para uno a cuatro flujos de vídeo.....	142

# Capítulo 1

## Introducción y objetivos

### 1.1. Introducción

Nadie quiere pensar en la posibilidad de verse involucrado en una situación de emergencia. Pero la mayor parte de las veces está fuera de nuestro alcance evitarlo. Lo que sí podemos hacer es trabajar para mejorar los mecanismos de actuación y procurar salvar el mayor número de vidas posible. En este contexto se sitúa la presente tesis doctoral.

La planificación de la gestión de crisis es esencial para ofrecer una rápida respuesta y para la adecuada toma de decisiones. Es muy importante formalizar y clarificar roles y responsabilidades, así como preparar bien los procedimientos y procesos que serán lanzados como respuesta a una emergencia.

El propósito de la gestión de emergencias es claro: mantener a los ciudadanos tan seguros como sea posible antes, durante, y después de que suceda una emergencia, bien sea una catástrofe natural (p.e. la explosión del Volcán Nevado del Ruiz en Colombia, en 1985; o el

terremoto del año 2011 en Lorca, España) o causada por el hombre (p.e. un incendio provocado o un ataque terrorista).

Un aspecto fundamental para alcanzarlo es el entrenamiento de operativos a todos los niveles; desde la intervención hasta el mando y control operacional, pasando por el mando y control táctico. Igualmente, son primordiales la interoperabilidad y la armonización de procedimientos entre los distintos organismos o agencias que participan en la mitigación de una emergencia; y sólo se pueden conseguir mediante ejercicios de entrenamiento conjunto.

La aplicación de las Tecnologías de Información y Comunicaciones (TIC) en la gestión de grandes emergencias es actualmente uno de los temas de investigación y desarrollo más relevantes en el ámbito mundial. En particular, el desarrollo de Sistemas de Información para Mando y Control (C2IS) específicamente aplicables al entrenamiento en gestión de situaciones de emergencia, así como de sistemas en los que éstos puedan interoperar con aplicaciones heterogéneas como, por ejemplo, mundos virtuales<sup>1</sup>.

La aproximación tradicional se basa en costosos equipos especializados en simulación y de aplicación específica. Pero las capacidades de simulación en red actuales, utilizando equipos de proceso comerciales (COTS) no han sido explotadas

<sup>1</sup> Un mundo virtual es un entorno de simulación basado en ordenadores en red, diseñado para interactuar con las personas a través de avatares (representación gráfica 3D de los usuarios) y agentes inteligentes. El entorno simulado se basará en entornos reales y en él se aplicarán las normas del mundo real [JoPr10].

suficientemente. Más aún, la interacción entre el mundo real y mundos virtuales usando formatos de representación estandarizados para el intercambio de información y garantizando la interoperabilidad, es un área de investigación relativamente reciente y en la cual hay todavía mucho por desarrollar.

El objetivo principal de la tesis, desde el punto de vista tecnológico, es la integración de un sistema de mando y control dentro de un sistema de simulación en red para entrenamiento basado en mundos virtuales y siguiendo el estándar MPEG-V [Inte13]. Particularmente en la implementación de un middleware para interconectar aplicaciones utilizando un formato intermedio de representación estandarizado. Todo esto supone un gran aporte al campo de la investigación y desarrollo en sistemas de información y comunicaciones para gestión de emergencias.

Las tres hipótesis de partida para desarrollar un sistema de estas características son:

1. Los sistemas tradicionales de entrenamiento basados en simulación son costosos y poco flexibles.
2. El entrenamiento real en el ámbito de la gestión de emergencias es muy caro y complicado en cuanto a la armonización de procedimientos entre agencias.
3. Un sistema que integre realidad y simulación en red, será por una parte, relativamente barato y, por otra, facilitará la interoperabilidad y la armonización de procedimientos entre agencias.

Hay que destacar que la tesis doctoral se aproxima a la generación de mundos virtuales como usuario. No se plantea una investigación en el campo de desarrollo de la realidad virtual, sino en el campo de la interconexión de sistemas de información para mando y control y redes de sensores con sistemas de simulación, basados en mundos virtuales, orientados al entrenamiento en la gestión de emergencias.

Con un continuo entrenamiento en un entorno como el propuesto, el personal que interviene en la resolución de crisis reales estará más familiarizado con la planificación de cada tipo de crisis así como con los procedimientos, roles y responsabilidades que les corresponden; y estará mejor preparado para actuar cuando se presente una emergencia, reduciendo así el tiempo de respuesta y mejorando la eficacia de actuación de las unidades desplegadas al poner en práctica los planes previamente entrenados.

La presente tesis contribuye sin duda a la implantación y extensión del estándar ISO 23005 (MPEG-V) en el campo de aplicación de las TIC a la gestión de emergencias, particularmente en el entrenamiento, la interoperabilidad y la armonización de procedimientos.

La tesis ha sido desarrollada en el marco del proyecto de investigación del Plan Nacional de I+D titulado “Sistema de entrenamiento C4ISR multimedia para gestión de emergencias, basado en la interconexión del mundo real y mundos virtuales” y con código TIN2010-18372.

## 1.2. Objetivos

El objetivo principal de la tesis es la integración de un sistema de mando y control en el campo de operaciones real dentro de un sistema de simulación en red para entrenamiento basado en la utilización de mundos virtuales. Particularmente, en la implementación de la **Pasarela MR-MV** que interconectará aplicaciones del mundo real y de mundo virtual, a través de un formato intermedio aplicando la normativa MPEG-V e implementando capas de adaptación y traducción de datos.

Se plantean los siguientes objetivos específicos:

1. Analizar la utilización de mundos virtuales para el entrenamiento y la planificación de la gestión de una crisis.
  - Llevar a cabo un análisis de los requerimientos de usuario para ajustar debidamente los eventos que pueden ser simulados.
  - Hacer una revisión bibliográfica sobre sistemas de mando y control para gestión de emergencias, así como del uso de mundos virtuales en actividades de entrenamiento para determinar los elementos a representar y consideraciones adicionales para el diseño del sistema.
2. Diseñar e implementar una arquitectura para interconectar sistemas de mando y control con mundos virtuales basada en el uso de equipos comerciales. La arquitectura debe ser coherente para una simulación distribuida en un entorno colaborativo.

- Definir una arquitectura en capas, modular, multiagente y multiusuario con avatares manejados por los participantes en el ejercicio.
  - Modelar el mundo virtual partiendo de las características del mundo real, a partir de un fichero vectorial manejado por el SIG del sistema de mando y control y del fichero de misión que recoge la estructura operativa y de sensores.
  - Crear un *middleware* para la interconexión del mundo real con el mundo virtual.
  - Conectar entidades y sensores reales con entidades virtuales utilizando el estándar MPEG-V para la traducción y adaptación de los datos transmitidos.
3. Implementar mecanismos de traducción y adaptación al estándar MPEG-V para proporcionar una solución estandarizada.
- Analizar el estándar ISO/IEC 23005 (MPEG-V) para determinar la sintaxis, semántica, y cualquier aspecto relevante que deba considerarse con el fin de lograr una solución consistente y que cumpla a cabalidad la normativa.
  - Definir el modelo de datos del sistema extendiendo los formatos de representación del estándar MPEG-V y particularizándolo para cada herramienta a integrar en el sistema como un módulo independiente.
  - Diseñar e implementar el mecanismo de comunicación entre módulos, basado en el uso de servicios web, incluyendo los



procedimientos adecuados para la traducción y adaptación de datos.

4. Contribuir a la armonización de metodologías y procedimientos en gestión de crisis.

- Mejorar la eficiencia en el intercambio de información y la coordinación entre las distintas agencias involucradas, reduciendo la falta de sincronización en las actuaciones debida a la falta de armonización de distintas metodologías.

5. Evaluar el sistema.

- Realizar mediciones de uso de recursos (CPU y ancho de banda) para determinar el rendimiento del sistema al implementar el *middleware*.

### **1.3. Estructura de la tesis**

La tesis consta de seis capítulos: en el capítulo 2 se desarrolla el estado del arte; el capítulo 3 detalla los antecedentes y consideraciones hechas para diseñar la arquitectura, cuya implementación es explicada en el capítulo 4. La validación de la arquitectura propuesta se define en el capítulo 5 y, finalmente, en el capítulo 6 se recogen los resultados obtenidos y las conclusiones de la investigación, además de señalar algunas posibles líneas de trabajo futuro.

## 1.4. Bibliografía

[Inte13] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO/IEC 23005, Information technology -- Media context and control. Parts 1-7*, 2013.

[JoPr10] JOVANOVA, BLAGICA y PREDÁ, MARIUS: Avatars interoperability in Virtual Worlds. En: *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*, 2010 — ISBN 9781424481125, pp. 263-268.

# Capítulo 2

## Estado del arte

### 2.1. Introducción

El primer paso de la investigación doctoral fue hacer una revisión bibliográfica exhaustiva para adquirir, y reforzar, los conceptos necesarios para complementar la formación previa y sentar una base de conocimiento sólida.

A lo largo de este capítulo se resumen los principales hallazgos, divididos en tres secciones: en la sección 2.2 hablaremos de los conceptos relativos a sistemas de mando y control y su uso en la gestión de emergencias. La sección 2.3 trata del uso de mundos virtuales en aplicaciones similares, características, limitaciones y aspectos clave; y por último, aunque no por ello menos relevante, en la sección 2.4 se analiza en profundidad el estándar MPEG-V para determinar el ámbito de aplicación con respecto a la arquitectura a implementar, las restricciones y ventajas de su aplicación.

## 2.2. Sistemas de mando y control

El concepto de **mando y control** tiene origen en el ámbito militar. Según el Departamento de Defensa de los Estados Unidos, se define como “el ejercicio de la autoridad y de la dirección del comandante apropiado sobre las fuerzas que tiene asignadas para el cumplimiento de una misión” [Arme14].

Los autores en [AlHa06] extienden esta definición y afirman que “el mando y control es el ejercicio de la autoridad y dirección por un comandante debidamente designado sobre las fuerzas adjuntas y asignadas para la consecución de una misión. Las funciones de comando y control se ejecutan a través de una estructura de personal, equipos, comunicaciones, instalaciones, y procedimientos empleados por un comandante en planear, dirigir, coordinar y controlar fuerzas y operaciones en la consecución de una misión”.

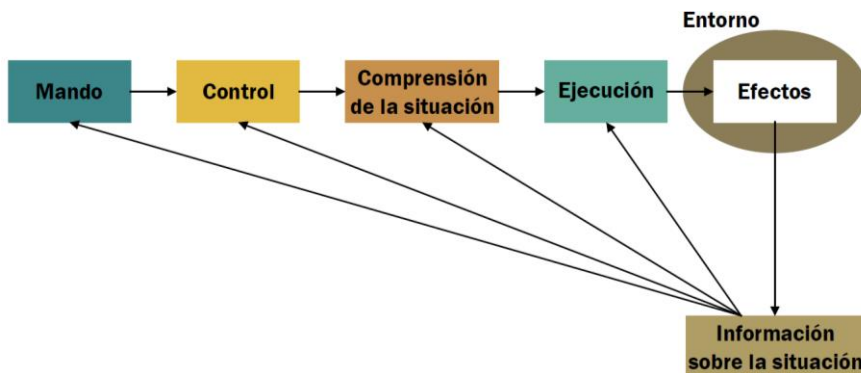


Figura 1. Modelo de un sistema de mando y control.

A partir de esto, se puede definir un sistema de información de mando y control (C2IS, *Command and Control Information System*)

como aquel en que existe una entidad que es responsable de gestionar los medios disponibles para cumplir un objetivo. Puede modelarse como en la Figura 1.

Podemos ver que los componentes del sistema son retroalimentados con la información de la situación; así, todas las decisiones serán tomadas con base en información actualizada en tiempo real.

Antes de proseguir, es importante aclarar que al hablar de “tiempo real” nos referimos a que existe un intervalo de tiempo definido dentro del cual los datos se consideran válidos. Y puede intuirse que éste es un aspecto clave en la gestión de emergencias, donde unos segundos pueden marcar la diferencia entre la vida y la muerte.

Retomando la Figura 1, veamos con más detalle cada componente:

- **Mando** es la definición de la situación inicial y la previsión de lo que podría ocurrir a futuro; se definen las intenciones de mando a partir de la información que se tiene.
- **Control** engloba la interpretación de las decisiones de mando, el seguimiento de la evolución de la situación, y la realización de los ajustes necesarios para mantener el sistema dentro de los márgenes definidos.
- **Comprensión de la situación**, o *situational awareness*, es la percepción compartida de la situación y su proyección a futuro. Es básica para la toma de decisiones y la definición de planes de acción.

- **Ejecución** son las acciones que se ejecutan y el tiempo en que se llevan a cabo.
- **Efectos**: modificación del entorno como resultado de la ejecución de una intención de mando.
- **Información sobre la situación**, a partir de la monitorización del entorno físico o cognitivo y de los efectos de la ejecución.

Podemos ilustrar la estructura de mando desde el punto de vista de actividades, según la estructura típica de un sistema de comando de incidentes (ICS, *Incident Command System*) que proponen los autores en [WTRM08] y que puede ser trasladada a los sistemas de mando y control. Ver Figura 2.

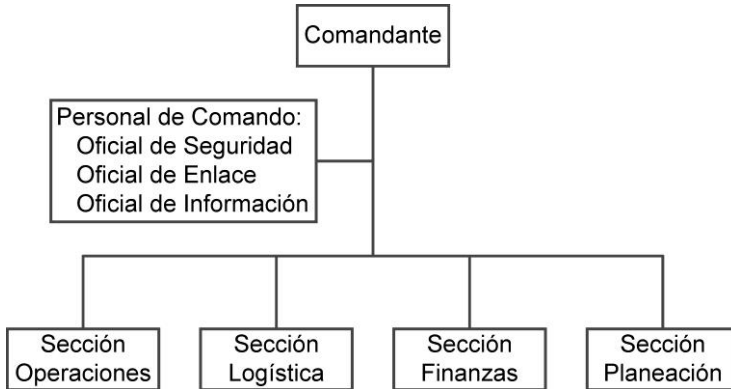


Figura 2. Estructura típica de un ICS.

En el caso de los C2IS, cada sección correspondería a un equipo de personal que corresponda (p. e. una brigada de bomberos o un equipo de paramédicos).

El mando y control ha ido evolucionando a la par con los avances

técnicos de forma que dichos avances condicionan nuevos enfoques y los procedimientos asociados. A su vez, las necesidades de mando y control generan nuevas soluciones técnicas en los ámbitos tecnológicos relacionados [Pére09].

Los sistemas C4ISR (*Command, Control, Computers Communications, Intelligence, Surveillance and Reconnaissance*) son la evolución de los C2IS. Comprenden un gran número de arquitecturas y sistemas de computación y de comunicaciones. Su principal objetivo es obtener información del teatro de operaciones y entregarla, convenientemente formateada, a aquellos al mando de una operación militar o de seguridad de manera que puedan construir una visión adecuada de la situación y tomar las decisiones correctas. Además, deben actuar como plataforma de comunicaciones para transmitir órdenes y cualquier otra información que pueda ser relevante en tiempo real [Pére09].

Existen múltiples aplicaciones y arquitecturas de mando y control, tanto en el ámbito civil como militar y en diversas áreas: operaciones militares, gestión de tráfico aéreo, gestión de operaciones espaciales, sistemas de detección y actuación para desastres naturales, operaciones en eventos de emergencia, e incluso en el entorno corporativo y la estructura de organizaciones [Ardi13].

Para el desarrollo de la presente tesis doctoral, el área que nos interesa, y en la que nos centraremos, es la de **gestión de emergencias**. La gestión de emergencias es el proceso mediante el cual individuos, grupos y comunidades gestionan los riesgos en un esfuerzo por evitar o disminuir el impacto de desastres resultantes en

caso de que el peligro sea inminente. Involucra cuatro fases: mitigación, preparación, respuesta, y recuperación [WaTR09].

En la fase de **mitigación** se busca evitar que los peligros se conviertan en desastres o, si lo hacen, reducir sus efectos. Los planes de acción son elaborados en la fase de **preparación** y ejecutados en la de **respuesta**. Finalmente, en la fase de **recuperación**, se busca restaurar el área a su estado previo.

La utilización de sistemas de mando y control en entrenamiento en gestión de emergencias permite que el personal se concentre en tareas críticas y de alto nivel (p. e., al tener la posibilidad de acceder a datos recogidos y pre-procesados en el sistema, el personal requerirá invertir menos recursos en el levantamiento de información). El uso de recursos se optimiza, la toma de decisiones es mejorada y todo esto se traduce en una reducción del número de fatalidades [MGWB03].

Gracias a los sistemas de mando y control para emergencias actualmente se puede estar mejor preparados para responder ante situaciones impredecibles, como catástrofes naturales o atentados, permitiendo salvar vidas y disminuyendo los daños. En el caso de sistemas C4ISR aplicados a las organizaciones las pérdidas se cifrarían en cantidades económicas y puestos de trabajo [Pére09].

Cuando ocurre una situación de emergencia los *first responders*, que son quienes primero asisten a la escena, tienen el primer contacto con la situación y toda la información recogida por ellos mediante sensores y otros dispositivos, es gestionada por el sistema de mando



y control para determinar el curso de acción según el punto de referencia (hacia dónde se quiere ir) definido. Por tanto, la ubicación de dispositivos [Pére09] y la visualización de información obtenida de cada unidad [FeGW00], son puntos clave para mejorar la **conciencia de la situación**.

La “**conciencia de la situación**” o *situational awareness* es un concepto fundamental en la teoría de los sistemas de mando y control, independientemente del ámbito en que se apliquen. Se refiere a la comprensión compartida de la situación e implica el análisis y retroalimentación de lo que está sucediendo; es indispensable para la adecuada toma de decisiones.

Otro punto clave es optimizar la coordinación de decisiones y el uso adecuado de recursos, pudiendo recurrir a modelos como el propuesto en [KSLN08] para representar la situación de desastre en términos de un problema de optimización y resolverlo con técnicas computarizadas. El uso de modelos complejos de optimización está fuera del alcance de esta investigación, pero es interesante comentarlo igualmente ya que podría ser una línea de trabajo futuro, así como lo sería el uso de redes sociales [AdSS12] como potenciador de mejora en protocolos de actuación, y para rastrear y mantenerse informado en casos de crisis o epidemias [Mert13].

Cuando se ha recogido la información necesaria para determinar el estado de la situación, se procede a ejecutar las acciones. Las unidades, dentro de lo posible, deben ser capaces de auto sincronizarse y tomar decisiones sin recurrir al nivel superior. Es vital entonces, que permanentemente el sistema se retroalimente en

tiempo real para incluir cualquier nueva información que pueda surgir. Se deben además considerar aspectos sociológicos y legales que también influyen en la gestión de una emergencia (medios, prensa, protección de datos, etc.).

Se hace evidente la importancia de filtrar adecuadamente la información existente para determinar una visión particular entre las múltiples que podían tomarse en consideración, y que se puedan tomar decisiones oportunas. Sin embargo, es de señalar que lo que se quiere ver o se quiere resaltar puede ser algo que no se sepa inicialmente o que vaya cambiando con el tiempo según evolucione el teatro de operaciones. Relacionado con este punto está el hecho de que los usuarios con mayor experiencia podrán formarse modelos con mayor rapidez y filtrar la información del entorno hacia modelos más adecuados a sus metas pero, como contrapartida, con una mayor predisposición a que las ideas preconcebidas y la falta de flexibilidad debido a la confianza en la experiencia conduzcan a *situational awareness* erróneas, esto es, no adecuadas a lo que la misión demanda respecto a sus objetivos [Pére09].

A nivel técnico [Pére09] se pueden esquematizar los sistemas C4ISR como un conjunto de  $N$  sensores,  $M$  actuadores, y  $O$  puestos de mando y control. Interconectándolos a todos, y como elemento fundamental, se encuentra una arquitectura de red. Los sistemas de mando y control para gestión de emergencias están basados en comunicaciones inalámbricas (se suelen usar ZigBee, WiFi, Mesh, multi-hopping, WiMAX y terminales satelitales) por lo que se suele disponer de poco ancho de banda y las tasas de errores pueden ser

altas. Según el nivel de control (red de campo, red táctica, red estratégica y red estratégica de coordinación) se utilizan más unas tecnologías u otras.

Es de vital importancia que los sistemas sean comprensibles, intuitivos y confiables [KSLN08] para facilitar su adopción y sacar verdadero provecho de su potencial para ayudar en el proceso de toma de decisiones y mejorar la coordinación y comunicación, tanto entre entidades de un mismo grupo como entre organizaciones. Otro aspecto crítico es la flexibilidad, pues el flujo de trabajo en tales entornos varía constantemente, creciendo o disminuyendo en función de los eventos que sucedan [WTRM08].

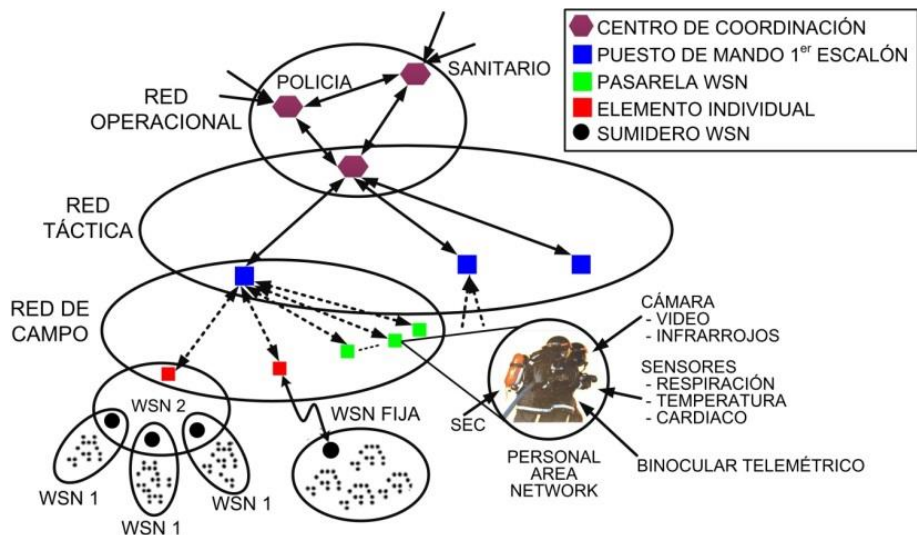


Figura 3. Arquitectura del sistema C4ISR SIMACOP.

Un ejemplo de sistemas C4ISR para gestión de emergencias es **SIMACOP**, Sistema de MAndo y COntrol de Pequeñas unidades, [EPHC09, EsPP13]. Este sistema es el resultado de dos proyectos

del Plan Nacional de I+D desarrollados por el Grupo de Sistemas y Aplicaciones de Tiempo Real Distribuido [Grup14]<sup>2</sup>. SIMACOP constituye un sistema de mando y control de nivel táctico y operacional, especialmente adecuado para la gestión de emergencias pues hace hincapié, en la parte operativa, en garantizar la seguridad de las personas involucradas en un primera intervención; y en la parte técnica, en la integración de sensores de todo tipo, incluyendo vídeo, así como en la integración de redes de datos inalámbricas (ZigBee, WiFi, MESH y WiMAX) como soporte de comunicaciones para el sistema de mando y control. Su arquitectura se muestra en la Figura 3.

SIMACOP ha constituido uno de los puntos de partida para el diseño del sistema y posterior implementación, pruebas y validación. Sobre esto volveremos en capítulos posteriores.

### 2.3. Mundos virtuales

Empezaremos definiendo tres términos que, si bien son diferentes entre sí, suelen usarse a la par e incluso, a veces, indistintamente: mundo virtual, realidad virtual, y realidad aumentada.

Un **mundo virtual** (MV) es un entorno simulado, basado en el mundo real (MR), en el cual los usuarios interactúan a través de avatares y agentes inteligentes [APEP15]. Cuando hablamos de **avatar** nos

<sup>2</sup> Proyectos TIN 2004-03588 "Sistema C4ISR multimedia para gestión de emergencias" y TIN 2007-67410 "Integración de redes de sensores en un sistema C4ISR multimedia para gestión de emergencias"

referimos a la representación gráfica de un usuario, usualmente 3D.

La **realidad virtual** es, según la Real Academia Española [Raer15], “la representación de escenas o imágenes de objetos producida por un sistema informático, que da la sensación de su existencia real”. Se intenta **sustituir** el mundo real a través de dispositivos que permiten al usuario “sentir” que se encuentra en otro lugar; sumergirse a una realidad que ha sido construida pero que no necesariamente existe en la realidad.

Por su parte, con la realidad aumentada lo que se pretende es “**mejorar**” la realidad; el usuario tiene una serie de información que aumenta sus sensaciones. Es decir que la realidad aumentada no reemplaza a la realidad sino que es un suplemento de ella; se agrega una capa adicional de información virtual sobre la información real ya existente mediante el uso de los dispositivos disponibles en el entorno del mundo real del usuario (p.e. teléfonos móviles o las *Google Glass* [Goog14]), aumentando sus sensaciones [KiRa13].

Tanto la realidad virtual como la realidad aumentada tienen varias fuentes de información y programación para generar simulaciones visuales, u orientadas hacia otros sentidos (p.e. olfato), y crear una experiencia sensorial. Por tanto, aunque ambas pueden generar sensaciones similares en el usuario, hay diferencias claras entre ellas; la más importante es que la realidad aumentada tiene lugar en el mundo real mientras que la realidad virtual no [KiRa13].

Las investigaciones en mundos virtuales iniciaron hace ya varias décadas. En 1962 aparece Sensorama, el primer cine con tecnología

3D. Y es en 1965 cuando, a partir de un artículo de Iván Sutherland, toma forma el concepto de realidad virtual; en él, Sutherland afirma que “una pantalla conectada a un ordenador digital nos da la posibilidad de familiarizarnos con conceptos no realizables en el mundo físico. Es una pantalla a través de la cual se puede ver un país de las maravillas matemático” [Suth65].

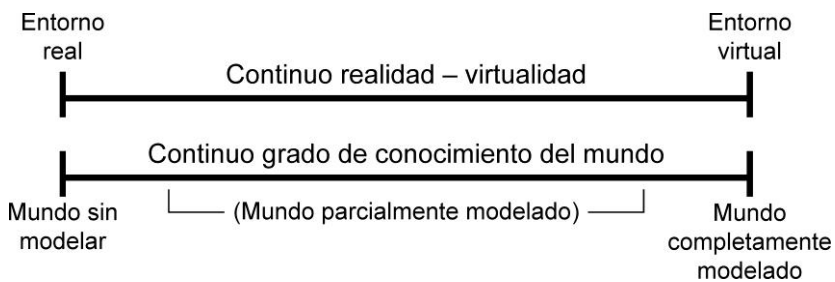
Sutherland, además, inventó el primer casco de realidad virtual y el primer sistema aplicado (llamado “*The sword of Damocles*”, “La espada de Damocles”); también desarrolló algoritmos para hacer más eficiente la utilización de gráficos 3D a partir del primer algoritmo para eliminar superficies oscuras y ocultas de una imagen, escrito por su compañero de estudios Lawrence Roberts.

En un principio el uso de mundos virtuales estaba orientado principalmente a juegos y entretenimiento (si bien en los años 70 e inicios de los 80 ya se fabricaban los primeros simuladores de vuelo para el ámbito militar). A día de hoy su utilización va mucho más allá [KoMF09, RiJo00], ya que se han convertido en una herramienta muy útil para mejorar las capacidades de aplicaciones del mundo real y han sido utilizados ampliamente para entrenamiento y aprendizaje en diversas áreas (medicina, educación, industria, juegos, etc.). Los usuarios pueden aprender y practicar la realización de tareas, a la vez que se desarrollan agentes con los que pueden actuar de manera colaborativa dentro del MV [RiJo00].

Al referirse a los sistemas que mezclan porciones del mundo real, con partes de uno o varios mundos virtuales, se suele hablar de realidad aumentada o realidad virtual (recordemos las definiciones

introducidas al inicio de la sección). Pero, existen otros términos que hacen distinción de la proporción de elementos virtuales o reales y que, si bien no comentaremos en profundidad porque están fuera del objeto de estudio de la tesis, resulta interesante exponer brevemente.

Paul Milgram y Fumio Kishino observaron [MiCo99] que se usaba el término “realidad aumentada” en diversos contextos: (i) sistemas que usan cascos o gafas de realidad virtual; (ii) sistemas que apuestan por el desarrollo gráfico y usan otros dispositivos (usualmente monitores) para aumentar la realidad; e incluso a veces a (iii) cualquier sistema que mezcle entornos virtuales y reales. Proponen entonces utilizar el término “**virtualidad aumentada**” para aquellos mundos predominantemente virtuales a los que se agregan elementos reales; y con “**realidad aumentada**” – coincidiendo con la definición hecha anteriormente – se refieren a aquellos sistemas reales sobre los que se añaden componentes virtuales. Finalmente, emplean el término “**realidad mixta**” para englobar los sistemas que mezclan parte real (realidad) y virtual (virtualidad).



**Figura 4. Continuo realidad – virtualidad en paralelo con el continuo grado de conocimiento del mundo.**

Otro concepto interesante, propuesto por los mismos autores, es el

de “**Continuo de realidad – virtualidad**” (*Reality – Virtuality continuum*). Sirve para describir un eje imaginario en el que se va pasando desde lo completamente real (extremo izquierdo) hacia lo completamente virtual (extremo derecho), y viceversa, abarcando todas las posibles variaciones y composiciones de objetos virtuales y reales (todos los sistemas en medio entrarían en la categoría de realidad mixta). Este continuo es paralelo al “**Continuo de grado de conocimiento del mundo**” (*Extent of World Knowledge continuum*), que hace referencia al nivel de conocimiento que tiene el ordenador con respecto al mundo que se presenta y si éste está completa, parcial, o mínimamente modelado. Ver Figura 4 [MiCo99].

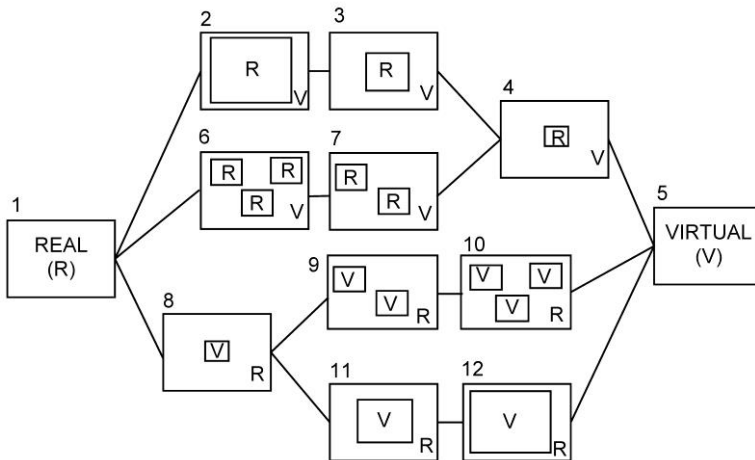
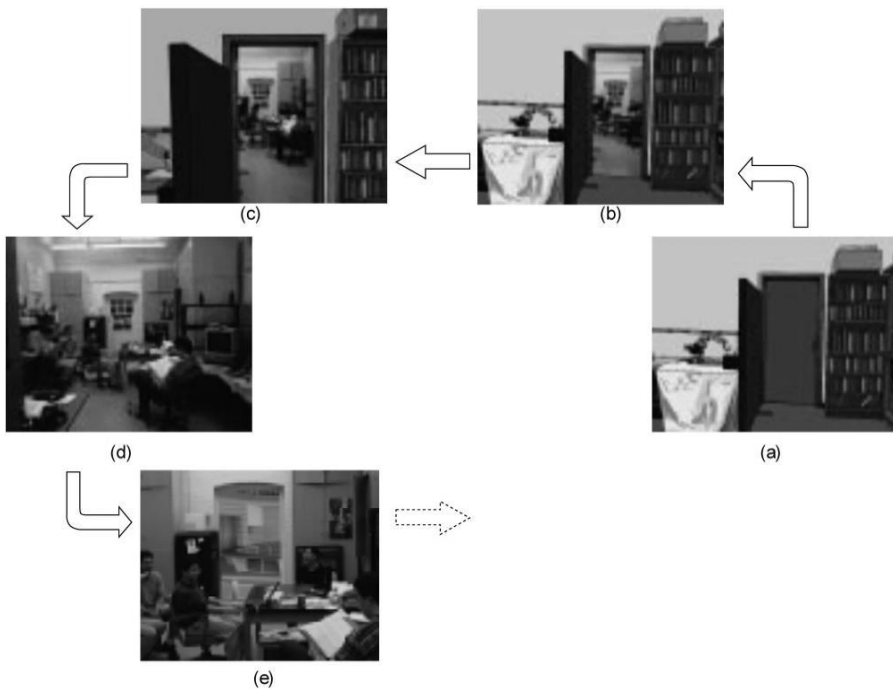


Figura 5. Ejemplo de espacio combinado de realidad mixta.

Las Figuras 5 y 6 muestran ejemplos de recorridos a lo largo del continuo realidad – virtualidad. La Figura 5 ilustra un ejemplo de espacio de casos de realidad mixta [MiCH99] en un simulador para recorrer en bicicleta en el Central Park de Nueva York [CaEE98]. Cada nodo tiene una proporción distinta de elementos virtuales y



reales. Al tomar, por ejemplo, el camino 5-4-3-2-1-8, se parte de un mundo completamente simulado y la proporción de objetos virtuales va disminuyendo a medida que se avanza, hasta llegar al mundo real y volviendo, a continuación, a un nodo con una porción virtual (el continuo realidad – virtualidad tiene una naturaleza circular).



**Figura 6. Ejemplo de recorrido de un mundo virtual al mundo real.**

La Figura 6 muestra un ejemplo similar, pero de manera más visual. Ilustra cómo sería el recorrido a lo largo del continuo realidad – virtualidad partiendo de (a) un entorno completamente virtual donde, al abrir la puerta, se ve (b) una porción de mundo real que (c) aumenta en el siguiente cuadro. En (d) se ha entrado a un entorno enteramente real y en (e) nuevamente aparece un componente

virtual, mostrando que no es necesario seguir la misma trayectoria para retornar al mundo virtual y demostrando, una vez más, la naturaleza circular del continuo realidad – virtualidad.

Resulta evidente que las prestaciones de los sistemas virtuales tradicionales pueden ser mejoradas y extendidas con la implementación de un sistema híbrido incluyendo datos de fuentes reales [AKAA12, VeSF11]. En el caso de gestión de emergencias, el uso de realidad virtual para crear ejercicios de entrenamiento permite al personal operar con modernos equipos informáticos, responder rápidamente ante eventos imprevistos en situaciones de estrés, y realizar ejercicios conjuntos con una reducción significativa en complejidad y costos.

En [MGWB03] los autores proponen un sistema con el cual un *first responder* puede hacerse una visión de la situación optimizando tiempo y recursos, y sin ponerse en riesgo prematuramente. Afirman que un sistema de estas características debe permitir:

- Obtener información de la víctima
- Analizarla para extraer la información relevante
- Difundir dicha información de la forma más eficiente posible

Con lo que el diseño del sistema parte de dos elementos base: los componentes de cada operativo (tales como sensores y equipos), y los componentes que conectan a los operativos (enrutamiento ad-hoc, mensajes).

Una decisión importante en las etapas preliminares de la investigación fue escoger una herramienta que, además de funcionar como generador de mundos virtuales, sirviese de punto de partida de diseño y posteriormente utilizada para la implementación. Se eligió OLIVE (*On-Line Interactive Virtual Environment*), una plataforma de software propietaria de la empresa estadounidense SAIC que proporciona capacidades de comunicación interactiva multimedia para colaboración, entrenamiento (ver Figura 7), operaciones y educación [Saic14]. Fue escogida por constituir uno de los entornos de mundos virtuales más avanzados del mercado y debido a sus prestaciones, que incluyen la posibilidad de conectar aplicaciones externas. Se comentará más al respecto en la sección 3.2.1.



Figura 7. Entrenamiento de *first responders* en OLIVE.

Una vez determinado el sistema de mando y control y la herramienta de generación de mundos virtuales que serían la base del diseño y posterior implementación, pruebas y validación; fue necesario estudiar y analizar los aspectos a considerar cuando se diseñan

aplicaciones en las que hay interacción con un entorno virtual, como se propone en esta tesis. Hablaremos sobre los principales hallazgos en los numerales 2.3.1 a 2.3.5.

### **2.3.1. Representación de entornos virtuales**

Uno de los aspectos claves al diseñar aplicaciones que usan mundos virtuales es la representación de elementos; es vital mantener la consistencia y coherencia con el mundo real y un nivel de realismo adecuado.

Existen numerosos estudios [AKAA12, HHKK13] en reconstrucción de escenas y usuarios del mundo real para crear una representación 3D en tiempo real que pueda ser situada en un mundo virtual compartido [BSCB08, HHKK13] e incluso con enfoque en información táctica [FeGW00], lo que contribuye a mejorar el *situational awareness*. También se ha estudiado la adición de información semántica [IbDe11].

Hasta hace algún tiempo, los modelos de representación de mundos virtuales se ocupaban de describir los mundos de manera que los navegadores pudiesen visualizar eficientemente la geometría y, en algunos casos, soportar interacciones de bajo nivel. A medida que se van requiriendo interacciones más ricas y a niveles más abstractos, surge la necesidad de un modelo de representación de alto nivel incluyendo descripciones semánticas de los objetos en el mundo, así como el desarrollo de agentes razonando sobre el entorno que habitan [IbDe11].

Así pues, los modelos han ido evolucionando para implementar mundos virtuales que soportan interacciones complejas, mejoran el aprendizaje de agentes e incluyen información semántica de alto nivel y ontologías [Grub95, IbDe11, PBTK05]. El uso de información semántica ha constituido una de las bases de diseño de la arquitectura propuesta en la presente tesis.

Utilizar un nivel semántico hace posible separar información abstracta (requerida para dominar la complejidad de una escena y su evolución dinámica) y concreta (representación visual directa del objeto que se desea controlar) de una aplicación. A su vez, hace posible crear modelos independientes de la aplicación [IbDe11]. Todo esto contribuye a aumentar la interoperabilidad y facilitar la interacción.

También es posible describir la semántica de una entidad a partir de ontologías. Tom Gruber al hablar de **ontologías** afirma que, en el ámbito de la inteligencia artificial, todo aquello que “existe” puede ser representado [Grub95]. Por otra parte, el diccionario Oxford la define como “una lista de conceptos y categorías en un tema que muestra las relaciones entre ellas” [Dicc15]. En esta tesis se adopta la acepción menos formal, pero más intuitiva, definiendo “ontología” como la estructura mediante la cual se describe cualquier tipo de entidad, sus atributos y relaciones.

El uso de ontologías permite a los agentes computacionales interactuar con el entorno virtual y razonar sobre él [EnTh11]. Resulta lógico entonces que una tendencia recurrente sea utilizar ontologías para modelar la información semántica de entornos virtuales. Dicha

información debe ser incluida durante la creación del mundo virtual [IbDe11] y mejora la habilidad de interacción con el mundo.

La definición de ontologías, cabe anotar, es una tarea compleja y que no debería ser subestimada. Es necesaria la colaboración de un experto para realizar las definiciones adecuadas y para implementarlas como una capa semántica de entornos virtuales. La gran ventaja es que dicha capa podría ser reusable por múltiples entornos y ayuda a promover la interoperabilidad.

Cabe señalar que no está dentro del alcance de la presente tesis la aplicación de un modelo ontológico, sino que los esfuerzos se han concentrado en diseñar un modelo de anotación de metadatos adecuado para el sistema propuesto, y cumpliendo la normativa del estándar ISO/IEC 23005, de la cual hablaremos en la sección 2.4.

Con respecto a la anotación de elementos, se recomienda [IbDe11] y promueve el uso de un esquema simple de anotación de metadatos relacionado con los elementos contenidos en el entorno virtual, independiente de la aplicación y, de ser posible, empleando anotación automática. Esta es una tarea que, si bien es bastante más sencilla, contribuye sustancialmente a la usabilidad, modularidad y a la eficiencia en la transmisión y análisis sintáctico de datos.

### **2.3.2. Arquitecturas**

Es habitual el uso de arquitecturas modulares y distribuidas. Por ejemplo, en [RiJo00] proponen una arquitectura con componentes separados ejecutándose en paralelo y que se comunican

intercambiando mensajes a través de un despachador, incrementando la modularidad. Se ilustra en la Figura 8 [RiJo00].

También es usual la definición de una arquitectura con una capa semántica intermedia que funciona como interfaz entre los agentes y el mundo virtual. Este enfoque ofrece dos grandes ventajas [IbDe11]: por un lado, permite desacoplar la representación gráfica de la semántica, de manera que los agentes pueden interactuar directamente con la capa semántica con el fin de procesar las representaciones del mundo. Por otro lado, este tipo de arquitectura es reutilizable por diferentes aplicaciones y permite desacoplar la representación gráfica de la semántica; así, los agentes pueden interactuar directamente con la capa semántica y procesar la representación semántica del mundo virtual.

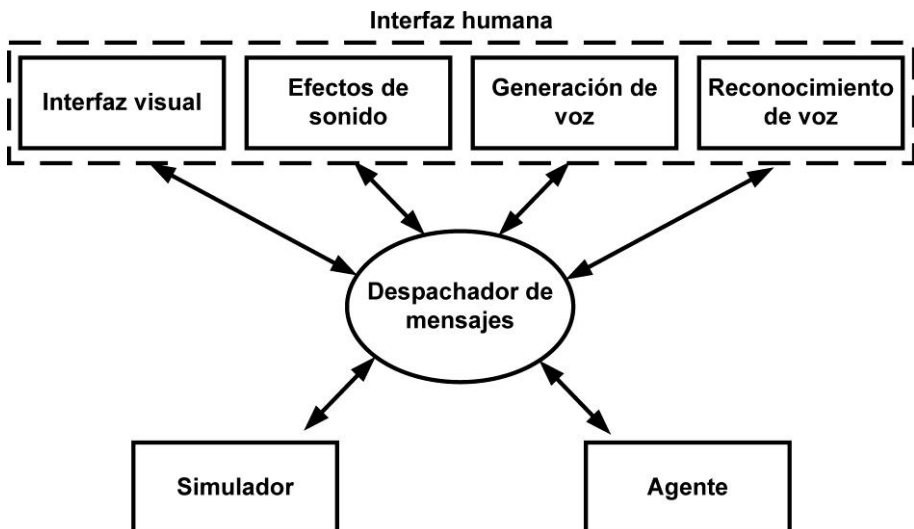


Figura 8. Ejemplo de arquitectura modular para mundos virtuales.

Sin embargo, el uso de arquitecturas yendo más allá de la

representación y buscando permitir la interacción directa entre el mundo real y el virtual no ha sido suficientemente explorado, menos aún si hablamos de un enfoque en gestión de emergencias. Esta es una de las brechas que se pretende cubrir con la tesis doctoral, al proponer una arquitectura en capas y con un *middleware* que conecta el sistema C4ISR operando en el mundo real con un mundo virtual representando el mismo campo de operaciones. Entraremos en detalle a este respecto en el capítulo 3.

### **2.3.3. Ayuda y soporte a navegación**

A menudo los usuarios experimentan dificultades y pueden desorientarse o perderse fácilmente por múltiples razones, debido tanto a problemas propios del mundo virtual, p. e. falta de puntos de referencia o nivel reducido de detalle [IbDe11], como a dificultades comunes con el mundo real (p. e. quedar atrapado dentro de un edificio). Para solventar estos problemas pueden utilizarse y combinarse varios métodos y técnicas dependiendo de la aplicación. Por ejemplo, los agentes animados pueden servir como guías de navegación previniendo que los usuarios se pierdan [RiJo00]; o ajustar el nivel de detalle mediante análisis de redes sociales para representar un avatar basado en sus interacciones con otras entidades [HHMS11].

Los autores en [IbDe11] profundizan en este tema y proporcionan un resumen de técnicas para soporte a la navegación.

### **2.3.4. Adición de vida a los entornos**



La adición de vida (poblar el entorno con agentes autónomos que se comporten de forma realista, tales como humanos y animales) ayuda a mejorar la experiencia de usuario y el realismo de la simulación [IbDe11]. Los autores en [EnTh11] han desarrollado aplicaciones semánticas que modelan el mundo real asociando ontologías con conceptos y ubicaciones.

En este punto resultaría especialmente útil el uso de realidad virtual y realidad aumentada. En el caso de un ejercicio de entrenamiento en gestión de emergencias, éste se vería enriquecido al agregar elementos sorpresa (p.e. personas heridas que aparecen de repente) y que ayudarían a analizar los protocolos de actuación.

Sin embargo, no debe verse como una tarea trivial. En [IbDe11] los autores hacen una serie de reflexiones y recomendaciones a tener en cuenta:

- La adición de nuevos comportamientos para aumentar el grado de realismo, incluidos aquellos que aumentan el rango de sociabilidad, debería ser priorizada por encima del refinamiento de los comportamientos ya existentes y procurando siempre mantener la consistencia.
- En cuanto a la agregación de personajes, hay que ser cuidadosos. Si es posible, se aconseja [IbDe11] evitar los humanoides ya que pueden generar expectativas equivocadas, especialmente en contextos interactivos, y recurrir al uso de personajes menos realísticos tales como robots, dibujos animados o animales, si caben dentro del contexto.

### **2.3.5. Interoperabilidad**

Por último, pero no menos importante, tenemos que hablar de interoperabilidad.

Un aspecto clave al diseñar nuevas aplicaciones en el que se pretende que interactúan diferentes sistemas es el uso de estándares [Egye07]. Si además nos centramos en el caso particular, y en el objetivo de implementar la interacción entre el mundo real y mundos virtuales aplicando formatos estandarizados y permitiendo el intercambio de datos entre mundos y usando equipos comerciales, el reto es aún mayor ya que es un área de investigación reciente y en la cual hay aún mucho por desarrollar.

Los esfuerzos para desarrollar estándares para mundos virtuales son relativamente recientes: iniciaron con el proyecto Metaverse 1 [Meta14] y sólo hasta el año 2011 se publicó un documento oficial por la ISO/IEC. Con el título “*Information technology — Media context and control*”, MPEG-V es un estándar emergente para interconectar mundos virtuales [Inte13, TGWH09] que proporciona la base para asegurar interoperabilidad a nivel de representación de datos [JoPr10]. Se comentará con detalle en la siguiente sección.

## **2.4. Estándar MPEG-V**

La interoperabilidad, como ya se ha comentado, es un aspecto clave a considerar en el diseño de implementaciones que pretendan interconectar aplicaciones, ya sean del mundo real o virtual, y es facilitada por el uso de estándares [Egye07].

Hasta hace relativamente poco, el desarrollo de estándares para mundos virtuales se orientaba principalmente a la representación gráfica de objetos. Es con el proyecto Metaverse1 [Meta14] que se inician los esfuerzos para crear un estándar para conectar mundos virtuales y el real, particularmente en la creación de un formato de representación estandarizado, conduciendo a la publicación en 2011 del estándar MPEG-V (ISO/IEC 23005, *Information Technology – Media context and control*). Su objetivo es ofrecer una base técnica sólida para aplicaciones y servicios multimedia inmersivos y multidimensionales [APEP13, Inte13].

MPEG-V define una arquitectura y un modelo datos para representar e integrar contenidos multisensoriales en entornos de usuario, haciendo posible la interoperabilidad a nivel de representación de datos [JoPr10] entre mundos virtuales y el mundo real y permitiendo reacciones simultáneas en ambos mundos ante cualquier cambio.

Se utiliza lenguaje XML (*eXtensible Markup Language*) para definir un esquema base que contiene un elemento raíz, del cual se desprenderá el conjunto de elementos necesarios para representar todas las entidades, junto con sus correspondientes atributos y valores posibles [Inte13]. Este esquema permite gran facilidad de desarrollo y adaptabilidad.

A partir de la normativa de sintaxis y semántica recogidas en el estándar se ha creado una definición genérica de datos que será utilizada en la implementación de la pasarela de interconexión (capítulo 4). Se utilizan referencias a los esquemas de definición de datos propios del estándar para los elementos que aplica.

Como se ha comentado anteriormente, el sistema de mando y control utilizado como base para el desarrollo de la tesis doctoral es SIMACOP. Y para la generación de mundos virtuales se ha usado la herramienta OLIVE. Sin embargo, ninguno de los dos fue diseñado bajo la normativa MPEG-V, por lo que ha sido necesario crear el modelo de datos equivalente para cada uno, con el fin de garantizar la interoperabilidad. Esto constituye una de las contribuciones más importantes de la tesis. Asimismo, se ha definido el modelo de datos correspondiente para la pasarela de interconexión en la que sistemas tanto de mundo real como mundos virtuales pueden interoperar, y un modelo de datos genérico que constituye la base para integrar nuevas herramientas al sistema.

#### **2.4.1. Términos, definiciones y abreviaturas**

A continuación se relacionan aquellos términos, definiciones y abreviaturas propios del estándar MPEG-V [Inte13] con mayor relevancia para el desarrollo de la tesis doctoral:

- **Adaptación RV (real a virtual):** Adaptación de los metadatos, capacidades y preferencias para generar comandos y/o información medida basada en los mismos. Se toman como entrada las capacidades del sensor y la información captada de los sensores y se adapta la información medida a las capacidades del sensor.
- **Adaptación VR (virtual a real):** Procesamiento de la información sensorial a utilizar dentro del contexto del mundo real.

- **Adaptación VV (virtual a virtual):** Adaptación de la representación nativa de la información relacionada con el mundo virtual (que se intercambiará con otro mundo virtual) al formato de representación estandarizado MPEG-V.
- **Área de estandarización A (Información de Control):** representación de la información de control desde y hacia dispositivos en el mundo físico y dentro y hacia el mundo virtual.
- **Área de estandarización B (Información Sensorial):** comprende la representación de la información (bidireccional) de los datos intercambiados entre el mundo físico y el virtual, así como el intercambio de información entre mundos virtuales.
- **Comandos de dispositivo:** adaptación de la representación nativa de información relacionada con el mundo real (a intercambiar con el mundo virtual) al formato de representación estandarizado de MPEG-V en ambas direcciones (de nativa a estandarizada y viceversa).
- **Dispositivo consumidor (consumer devices):** receptor de comandos sensoriales y fuente de capacidad sensorial del sensor (SDC).
- **Dispositivo sensorial (sensory device, SD):** cualquier dispositivo consumidor para el cual se puede crear el correspondiente efecto sensorial (p. ej. luces, ventiladores...).
- **Dispositivo de interacción:** aquel que recibe entradas de usuarios y/o da salidas a usuarios.

- **Dispositivo de mundo real S:** un dispositivo del mundo real que contiene un sensor (de posición, de temperatura, de intensidad de luz...).
- **Dispositivo de mundo real A:** dispositivo del mundo real que contiene un actuador (monitor, altavoz, robot...) o puede contener también una combinación de sensores y actuadores. En nuestro caso, un mensaje con órdenes también puede verse como un actuador ya que esas órdenes causarán algún efecto en el sistema.
- **Información medida:** información adquirida por un sensor; p.ej. posición de una unidad.
- **Motor de adaptación:** donde se hace la adaptación real a virtual y/o virtual a real.
- **Representación de datos del mundo real:** representación nativa de la información relacionada con el mundo real que se pretende intercambiar con el mundo real (bien sea importada o exportada).
- **Representación de datos del mundo virtual R:** representación nativa de la información relacionada con un mundo virtual que se quiere intercambiar con el mundo real (ya sea importada o exportada).
- **Representación de datos del mundo virtual V:** representación nativa de la información de un mundo virtual a intercambiar con otro mundo virtual (bien sea importada o exportada).

- **SEDL:** *Sensory Effects Description Language*. Lenguaje de descripción de efectos sensoriales.
- **SEM:** *Sensory Effect Metadata*. Metadatos de efectos sensoriales.
- **SEV:** *Sensory Effects Vocabulary*. Vocabulario de efectos sensoriales.
- **XSI:** *XML Streaming Instructions*. Atributo que define el espacio de nombres que va a ser utilizado en un esquema XML.
- **Sensor:** dispositivo consumidor del que se puede recoger información del medio o entradas (p. ej. sensor de temperatura, de distancia, etc.).

#### **2.4.2. Partes del estándar**

El estándar MPEG-V (ISO/IEC 23005) se divide en 7 partes:

- Parte 1: Arquitectura (23005-1).
- Parte 2: Información de control (23005-2).
- Parte 3: Información sensorial (23005-3).
- Parte 4: Características de objetos de mundo virtual (23005-4).
- Parte 5: Formato de datos para dispositivos de interacción (23005-5).
- Parte 6: Tipos y herramientas comunes (23005-6).

- Parte 7: Software de referencia y conformidad (23005-7).

Cada parte especifica consideraciones a tener en cuenta con respecto a tipo de datos, formato, etc., y va acompañada de los esquemas XML correspondientes. Profundizaremos en cada una a lo largo de los numerales 2.4.2.1 a 2.4.2.7.

### 2.4.2.1. Parte 1: Arquitectura

La Figura 9 [Inte13] muestra la arquitectura de MPEG-V para el control de objetos e intercambio de información dentro del mundo real, entre mundos virtuales, o entre mundo real y virtual.

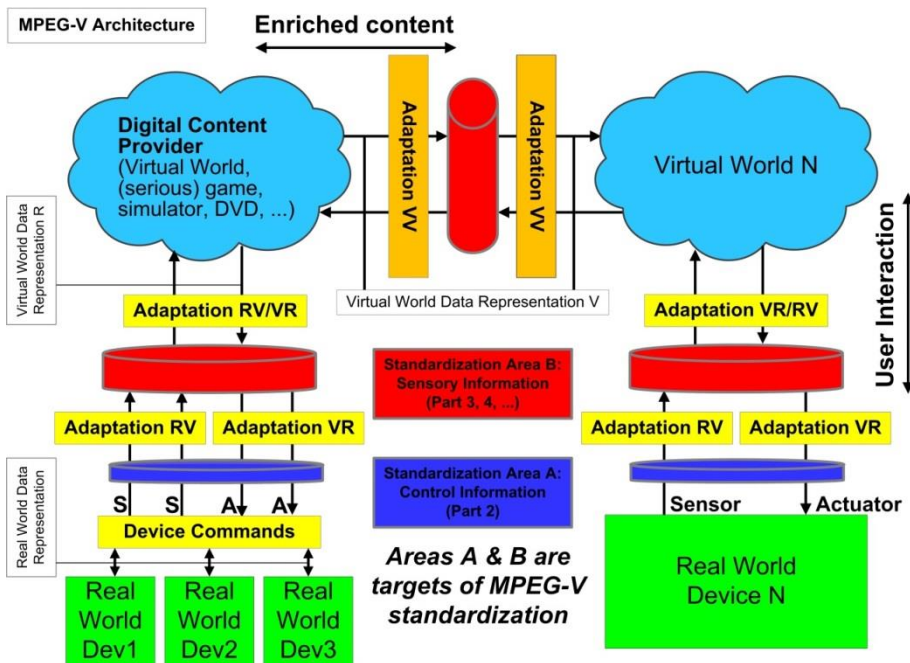


Figura 9. Arquitectura MPEG-V.

Esta arquitectura permite reacciones simultáneas en ambos mundos



a cambios en el medio ambiente y en el comportamiento humano, razón por la cual el estándar constituye una solución para integrar contenido multisensorial en entornos de usuario garantizando interoperabilidad al ofrecer un amplio abanico de herramientas para representar dicho contenido.

El intercambio de datos puede ser entre mundo real y mundo virtual o entre mundos virtuales. Adicionalmente, se contempla el control de avatares y objetos virtuales con señales del mundo real; y el control de objetos reales con señales del mundo virtual [Inte13].

Se quiere tener la capacidad tanto de emular situaciones reales en el mundo virtual como de tomar decisiones en el mundo virtual y que produzcan algún efecto en el mundo real. Por esta razón MPEG-V dentro de la arquitectura (Figura 9) define capas de adaptación **virtual a real (VR)** y real a **virtual (RV)**.

En el caso de adaptación real a virtual, se toma la información de los dispositivos de mundo real y se traduce en comandos, que pasarán por la capa de adaptación RV [Inte13]. Las salidas son recibidas por el proveedor de contenidos que a su vez los transmite a una capa de adaptación virtual a virtual VV, que permite la interconexión e intercambio de datos entre múltiples mundos virtuales.

Siguiendo un esquema similar, la adaptación virtual a real parte de la información recibida del proveedor de contenidos, que pasará por capas de adaptación VR para enviar comandos a los actuadores y obtener el efecto deseado en los dispositivos del mundo real.

Todo el proceso de conversión, adaptación y traducción se lleva a cabo en el **motor de adaptación**, y éste constituye el núcleo de lo que hemos denominado **pasarela de interconexión** (que, cabe recordar, es uno de los mayores aportes de la tesis doctoral) y que se irá explicando en detalle a lo largo de esta memoria.

Cabe señalar que MPEG-V no estandariza el motor de adaptación (ya sea RV o VR) en sí mismo; lo que define es un esquema que arroja como salida comandos a dispositivos sensoriales e información para controlar el mundo virtual [HHHK10].

Todo elemento del mundo real tendrá su equivalente representación en una entidad del mundo virtual. MPEG-V clasifica las entidades de un mundo virtual como **objetos virtuales** y **avatares**.

Se ha definido **avatar** como la representación gráfica de un usuario manejada por ordenador y mediante la cual existe e interactúa dentro del mundo virtual. En general, suelen ser representaciones humanas pero en esta tesis, y en consonancia con lo planteado en el estándar ISO/IEC 23005, los avatares podrán también ser vehículos, soldados, persona al mando (PC), etc. Los objetos virtuales, por su parte, son todos aquellos objetos propios de cada entorno y con los que no se puede ejecutar ninguna acción (como, por ejemplo, un edificio).

La apariencia y características propias de un avatar pueden ser definidas y compartidas entre todos los mundos virtuales en los que participa un usuario. Así, puede tener una identidad anónima pero común en cualquier mundo virtual que utilice [Inte13].

Sin embargo, la capacidad de transferir información entre mundos virtuales depende del grado de similitud entre ellos, con lo que nuevamente se evidencia la importancia de la estandarización y de implementar un formato intermedio común para poder transferir información entre ellos.

$n$  mundos virtuales pueden comunicarse entre sí, pasando la información por la capa de adaptación (recordemos que no está sujeta a estandarización) que interpreta y traduce los datos manteniendo la consistencia e integridad. Se puede consultar un ejemplo de aplicación en [Kane10], donde se propone e implementa un protocolo para intercambio de información entre mundos virtuales basado en PHP con XML y conexión a base de datos usando SQL.

Esta capa intermedia ha sido considerada dentro del diseño de la arquitectura propuesta previendo la comunicación tanto entre el mundo real y el virtual como entre mundos virtuales (independientemente de la herramienta con que hayan sido creados) con el fin de garantizar escalabilidad, adaptabilidad e interoperabilidad.

Otro concepto interesante es el de la **presencia social** dentro del mundo virtual [Inte13]. Desde hace varios años, se viene trabajando en el diseño del *Metaverso*, que es aquel espacio (universo) compartido en el que se supera la distancia física entre dos entidades en distintos lugares, al proporcionar intercambio de información entre ellos en tiempo real. La distribución de dicha información procesada en tiempo real crea una representación del usuario en varios lugares a la vez, permitiéndole utilizar aplicaciones,

dispositivos y servicios ubicados en equipos remotos.

Por otra parte, la comunicación entre mundos virtuales ofrece la posibilidad de **toma de decisiones en grupo** en el contexto de planeación espacial [Inte13]. Los mundos virtuales pueden servir como plataforma de comunicaciones a cualquier tipo de organización; por ejemplo, en la gestión de sistemas C4ISR para definir la ubicación de unidades militares en campo.

Finalmente, aunque no menos importante, se aplica el intercambio de información entre mundos virtuales para la **instanciación de objetos virtuales** [Inte13]. Cuando se tiene información de un objeto virtual ésta puede ser utilizada en diferentes mundos virtuales para crear y usar los objetos correspondientes de forma rápida y sencilla, heredando las características y capacidades. La información de los objetos virtuales se define mediante metadatos. Tanto avatares como otros objetos virtuales podrán ser controlados con señales del mundo real.

De manera similar, la información del mundo virtual se traduce (a través de las capas de adaptación VR) en comandos que se transmiten a los dispositivos del mundo real.

#### *2.4.2.2. Parte 2: Información de control.*

Proporciona herramientas para describir las capacidades de los dispositivos sensoriales y proveer interoperabilidad en control de dispositivos, así como mundos virtuales. Incluye descripciones de capacidad de los actuadores (dispositivos sensoriales) y sensores en

el mundo real así como las preferencias de usuario para generación de contenido y control de dispositivos. Ayuda a adaptar el control a cada caso de uso [Inte13].

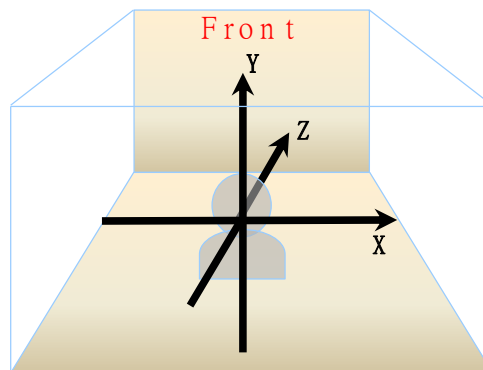


Figura 10. Sistema de coordenadas de referencia.

Se definen también tipos base para caracterizar las capacidades de dispositivos sensoriales (*SensoryDeviceCapabilityBaseType*) y de sensores (*SensorCapabilityBaseType*). La ubicación de los sensores se determinará según el mundo real del usuario y utilizando el sistema de coordenadas de la Figura 10.

Por otra parte, el tipo *UserSensoryPreferenceBaseType* permite definir las preferencias de usuarios individuales con respecto a la experiencia sensorial.

#### 2.4.2.3. Parte 3: Información sensorial.

Esta parte de MPEG-V se enfoca en la representación de la información de entrada a la capa de adaptación VR (que, recordemos, no está sujeta a estandarización); especifica la sintaxis y semántica para describir información y efectos sensoriales. Su

objetivo es mejorar la experiencia del usuario [KCPK11].

La intención básica de uso de la información sensorial en el estándar es mapear el contenido de las intenciones del autor a metadatos de efectos sensoriales y capacidades de dispositivo sensorial, como se muestra en la Figura 11 [Inte13] para reproducir algún efecto.

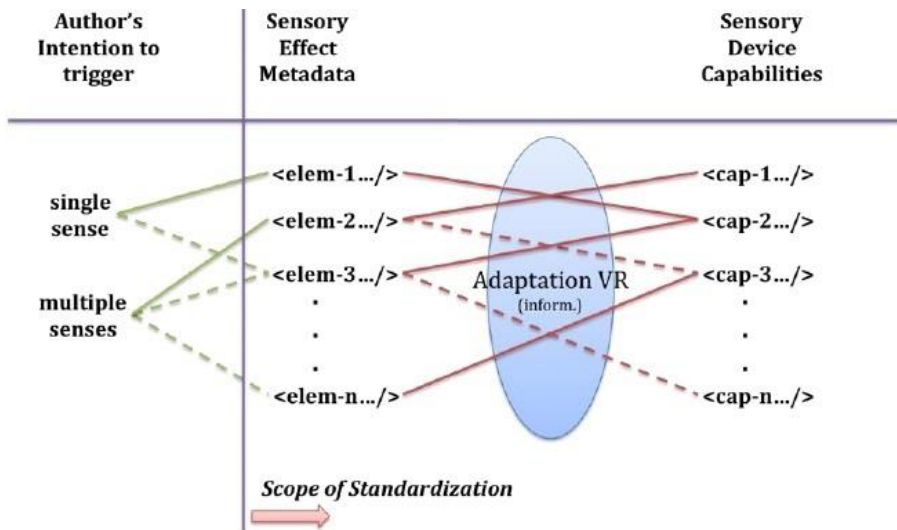


Figura 11. Mapeo de intenciones de mando.

Por lo general, un autor en este contexto trata de estimular uno o varios sentidos, para lo cual usará uno o varios elementos o tipos de datos para desencadenarlos. Por otro lado, los elementos SEM o tipos de datos son mapeados a comandos de control de dispositivos sensoriales. Esto, por lo general, se hace en la capa de adaptación VR, que no está sujeta a estandarización.

Nótese que el mapeo no necesariamente es uno-a-uno entre sentidos y elementos. Por ejemplo, la termo-recepción puede

producirse con un único efecto de temperatura, pero también puede combinarse con un efecto de viento.

El lenguaje de descripción de efectos sensoriales, SEDL (*Sensory Effect Description Language*) proporciona los bloques básicos de construcción para la creación de metadatos de efecto sensorial [Inte13]. Todo documento que describa la representación de efectos debería ser validado contra el esquema SEDL. Los detalles de sintaxis, semántica, y algunas reglas de validación adicionales se pueden consultar en el estándar.

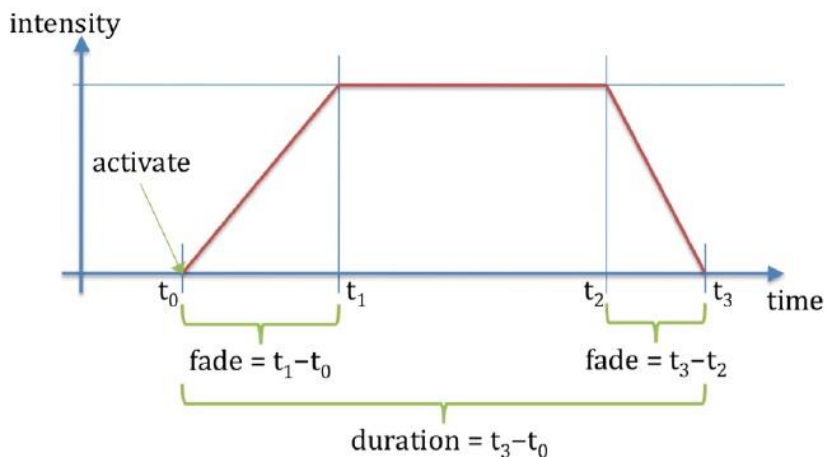


Figura 12. Modelo de tiempo para metadatos de efectos sensoriales.

Adicionalmente, MPEG-V adopta el modelo de tiempo ilustrado en la Figura 12 [Inte13] para metadatos de efectos sensoriales (útil, por ejemplo, para transmisión de vídeo).

Cada efecto puede ser activado ( $t_0$ ) y desactivado ( $t_3$ ) en ciertos puntos del tiempo. La activación de un efecto puede ser definida explícitamente (p. e. activar = *verdadero*) o indicada en función de la

duración de un atributo (p. e.  $t_3 - t_0$ ). Además, se pueden especificar tiempos de aparición ( $t_1 - t_0$ ) o desvanecimiento ( $t_3 - t_2$ ) durante los cuales el efecto debe corresponder a la intensidad definida.

#### 2.4.2.4. Parte 4: Características de objetos de mundo virtual.

Los objetos de mundos virtuales (OMV), como ya se comentó anteriormente, pueden ser clasificados en **avatares** y **genéricos**; y tienen dos objetivos: primero, caracterizar los objetos dentro del mundo virtual y, segundo, proporcionar interacción con el mundo virtual [Inte13].

Según el planteamiento del proyecto, será posible que la persona al mando del puesto de control (PC) cree su propio avatar, además de una unidad por cada subordinado bajo su mando; este sería el caso para entrenamiento de unidades de nivel superior. Otra posibilidad es que tanto unidades de mismo nivel como de niveles inferiores sean creadas por una persona del mundo real y que tales avatares los controle bien sea el usuario correspondiente o su superior en la cadena de mando. Esto variará según el despliegue deseado para el entrenamiento. Por otra parte, los avatares propios del mundo virtual también deben ser instanciados.

Los **objetos genéricos** permiten caracterizar objetos e interactuar con ellos [Inte13]. Podría decirse que son aquellos que hacen posible la ejecución de una acción, o son consecuencia de un evento, p.e., coches, vídeo, alarmas, mensajes, sensores...



Es necesario ser capaz de crear un objeto una vez e importarlo/usarlo en diferentes entornos virtuales. Existen tres requerimientos principales a tener en cuenta [Inte13]:

- Debe ser posible crear fácilmente importadores/exportadores desde varias implementaciones de entorno virtual.
- Controlar un objeto debe ser fácil.
- Debe ser posible modificar una plantilla propietaria (específica del mundo virtual) del objeto usando datos contenidos en el archivo de definición de datos.

Adicionalmente, debería ser posible controlar el objeto desde aplicaciones o recursos externos.

#### *2.4.2.5. Parte 5: Formato de datos para dispositivos de interacción.*

Esta parte del estándar contiene las herramientas para intercambiar información entre dispositivos de interacción; específicamente, define la normativa de formatos de comandos para control de dispositivos sensoriales y formatos de datos para recibir información de sensores.

Tiene como objetivo proporcionar modelos de datos para los dispositivos de interacción usualmente utilizados en la industria: sensores y actuadores. Estos formatos de datos pueden ser utilizados por variadas aplicaciones que soportan diferentes tecnologías MPEG, como, por ejemplo, ISO/IEC 23007 (MPEG-U, estándar para interfaces de *widgets* [Inte10]) e ISO/IEC 14496-20 (especificaciones para representación de escenas [Inte08]).

Pueden darse dos casos [Inte13] al controlar un mundo virtual usando herramientas MPEG:

- Cuando el mundo virtual usa una escena definida con herramientas MPEG (BIFS, Laser, etc.) los sensores y actuadores pueden ser conectados directamente a él mediante una interfaz MPEG-U.
- Cuando el mundo virtual ha sido creado con otras herramientas (tal como OLIVE, por ejemplo), se requiere un motor de adaptación y traducción.

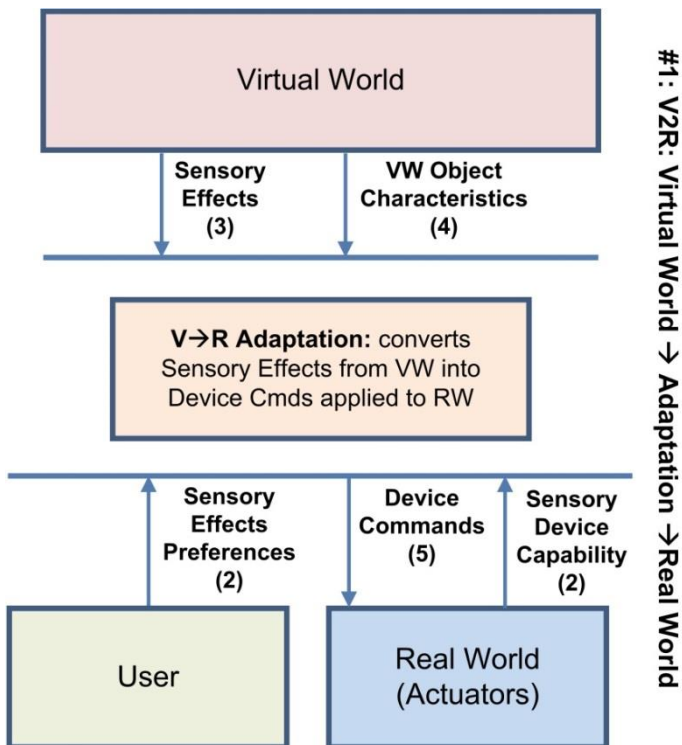


Figura 13. Alcance de MPEG-V parte 5. Caso 1: adaptación virtual a real.

La Figura 13 [Inte13] ilustra el alcance del modelo de datos para dispositivos de interacción para el caso de adaptación virtual a real. La adaptación VR toma efectos sensoriales (parte 3 del estándar) de un mundo virtual, opcionalmente capacidades sensoriales de dispositivo (parte 2 del estándar) de los actuadores y las preferencias de efectos sensoriales (parte 2 nuevamente) de los usuarios; genera comandos de dispositivo al adaptar los efectos sensoriales con base en las capacidades y/o preferencias.

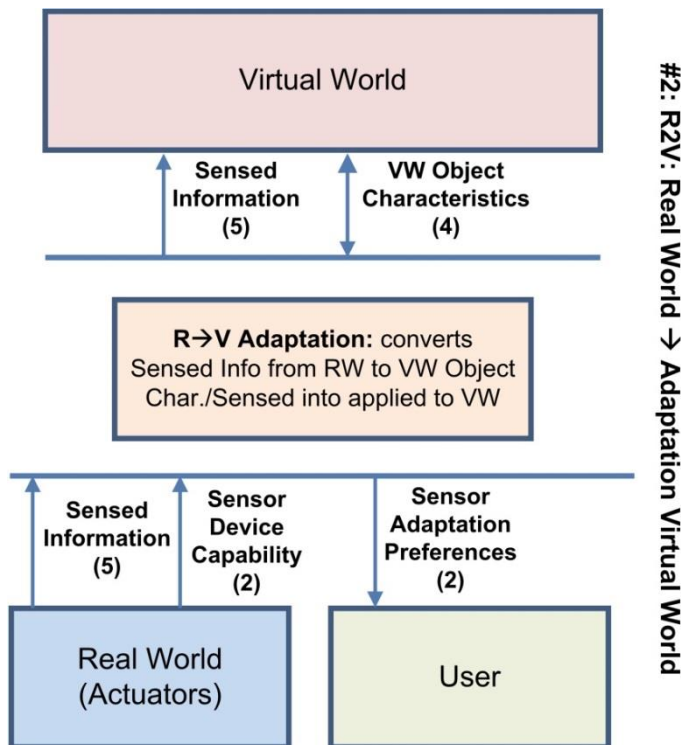


Figura 14. Alcance de MPEG-V parte 5. Caso 2: adaptación real a virtual.

La Figura 14 [Inte13] muestra el alcance del modelo de datos para dispositivos de interacción para el caso inverso, adaptación virtual a real. Se toma la información adquirida y capacidades sensoriales

(parte 2 del estándar), si están definidas, de los sensores y las preferencias de adaptación sensorial (parte 2 de nuevo) de los usuario, y/o las características de objetos de mundo virtual (parte 4 del estándar) del mundo virtual. Tiene dos usos: controla las características de objetos de mundo virtual, o transmite una versión adaptada de la información medida.

Los tipos base, punto de partida para capturar y controlar el mundo real [Inte13], para definir comandos de dispositivo y describir la información detectada son *DeviceCommandBaseType* y *SensedInfoBaseType*, respectivamente. Las instancias de posterior información detectada pueden ser generadas como una salida de los sensores.

#### 2.4.2.6. Parte 6: Tipos y herramientas comunes.

Se definen los tipos de datos y herramientas comunes a más de una herramienta definida en más de una parte del estándar [Inte13]. Contiene la definición del esquema base en XML y la normativa para declaración de variables.

Define además los siguientes tipos de datos básicos:

- *unitType*: descripción de una unidad como referencia al esquema de clasificación determinado por *UnitTypeCS*, definido en el Anexo A, sección A.2.1 de ISO/IEC 23005-6.
- *InclineAngleType*: ángulo de inclinación (de  $-360^{\circ}$  a  $360^{\circ}$ ).
- *Float3DVectorType*: vector con los valores para la posición

$(x, y, z)$ .

- $X, Y, Z$ : valor flotante (puede ser fuerza, torque, posición) para los ejes  $X, Y$  y  $Z$  respectivamente.

#### 2.4.2.7. Parte 7: Software de referencia y conformidad

Esta parte del estándar ISO/IEC 23005 especifica el software de referencia y conformidad implementando las cláusulas normativas de todas las partes del estándar. Esto se puede aplicar para determinar los módulos de software de referencia disponibles para cada parte entendiendo y utilizando los módulos disponibles. Los módulos de referencia disponibles son especificados como interfaces de programación de aplicaciones (APIs) de acuerdo a ISO/IEC 23006-1:2011 (MPEG *extensible middleware* (MXM) -- *Part 1: MXM architecture and technologies*).

Se consiguen tres propósitos [Inte13]:

- Validación de las especificaciones definidas en cada parte del estándar.
- Clarificación de las especificaciones de todas las partes de MPEG-V.
- Pruebas de conformidad para comprobar la interoperabilidad de las distintas aplicaciones contra el software de referencia que pretende cumplir con ISO/IEC 23005.

En definitiva, esta parte del estándar MPEG-V es la que define el

esquema contra el cual se validará el modelo de datos implementado en la pasarela de interconexión.

### **2.4.3. Aplicaciones y ejemplos**

Existen múltiples escenarios de uso dependiendo del tipo de intercambio de datos, que puede ser: (i) adaptación de información del mundo real al virtual, (ii) adaptación de información del mundo virtual al real e (iii) intercambio de datos entre mundos virtuales [Inte13]. Por otra parte, el estándar propone varias instanciaciones; a continuación se comentan algunos ejemplos de aplicaciones desarrolladas con base en estas instanciaciones.

**Broadcast 4D:** es posible crear y distribuir metadatos de efectos sensoriales. Tras los correspondientes procesos de binarización, multiplexación, y demultiplexación, el receptor extrae los metadatos para configurar dispositivos y generar comandos. Tales comandos de dispositivo, definidos por la normativa MPEG-V, son entregados a los dispositivos encargados de reproducir los efectos sensoriales. Un ejemplo de implementación de este caso es presentado en [YCLL10, Yoon13] para entretenimiento casero; mientras que los autores en [ChPa13] proponen un método para generar imágenes 4D con una aplicación para teléfonos inteligentes. Finalmente, en [KLKR13] se introduce un sistema para transmisión de datos basado en información háptica.

**Interacción natural con sensores inteligentes:** este es un ejemplo de cómo sacar provecho del potencial de las cámaras más allá de limitarse a capturar vídeo, usándolos además como dispositivos de

interacción. Es el caso de [HHHK10], donde los autores estudian el uso de MPEG-V para controlar objetos virtuales usando dispositivos del mundo real, enfocándose en la animación facial de avatares controlando la expresión mediante movimientos de la cara.

### **Interacción transparente entre el mundo real y mundos virtuales:**

Uno de los objetivos buscados al conectar mundo real y virtuales es alcanzar la interoperabilidad de manera transparente. En [KJSH13] proponen un método para transformar la información adquirida en el mundo real a datos con formato MPEG-V para controlar objetos de mundo virtual. La presente tesis doctoral también es un ejemplo de este tipo de instanciación; como se ha comentado repetidas veces ya, garantizar la interoperabilidad ha sido uno de los fundamentos de diseño de la arquitectura propuesta.

## **2.5. Bibliografía**

[AdSS12] ADAM, N R; SHAFIQ, B; and STAFFIN, R: Spatial Computing and Social Media in the Context of Disaster Management. In: *Intelligent Systems, IEEE* vol. 27 (2012), Nr. 6, pp. 90–96 — ISBN 1541-1672 VO - 27.

[AKAA12] ALEXIADIS, D S; KORDELAS, G; APOSTOLAKIS, K C; AGAPITO, J D; ET AL.: Reconstruction for 3D immersive virtual environments. In: *Image Analysis for Multimedia Interactive Services (WIAMIS), 2012 13th International Workshop, 2012*, pp. 1–4.

[AlHa06] ALBERTS, DAVID S. and HAYES, RICHARD E.:

*Understanding Command and Control*. 1st. ed. : CCRP Publications, 2006 — ISBN 1893723178.

[APEP13] ARDILA, LAURA; PÉREZ-LLOPIS, ISRAEL; ESTEVE, MANUEL; and PALAU, CARLOS E.: LVC Training Environment for Strategic and Tactical Emergency Operations. In: *10th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2013)*. Baden-Baden, Germany, 2013 — ISBN 9783923704804, pp. 140–144.

[APEP15] ARDILA, LAURA; PÉREZ-LLOPIS, ISRAEL; ESTEVE, MANUEL; and PALAU, CARLOS E.: Interoperable architecture for joint real/virtual training in emergency management using the MPEG-V standard. In: *Computer Standards & Interfaces* vol. 41 (2015), pp. 39–55.

[Ardi13] ARDILA, LAURA: *Pasarela de interconexión entre sistemas C4ISR y mundos virtuales aplicando el estándar MPEG-V*, Tesina de Máster, Máster Universitario en Tecnologías, Sistemas y Redes de Comunicaciones, Universitat Politècnica de València, 2013.

[Arme14] ARMED FORCES OF THE UNITED STATES: *JP 1-02. Department of Defense Dictionary of Military and Associated Terms*. vol. 2010, 2014.

[BSCB08] BOULOS, MAGED N KAMEL; SCOTCH, MATTHEW; CHEUNG, KEI-HOI; and BURDEN, DAVID: Web GIS in practice VI: a demo playlist of geo-mashups for public health neogeographers. In: *International*



*Journal of Health Geographics* vol. 7 (2008), Nr. 1, p. 38.

[CaEE98] CARRARO, GIANPAOLO U.; EDMARK, JOHN T.; and ENSOR, J. ROBERT: Techniques for handling video in virtual environments. In: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*, 1998 — ISBN 0897919998, pp. 353–360.

[ChPa13] CHOI, SEONG GON and PARK, MI RYONG: An Architecture and Method using MPEG-V Metadata in Smartphone and Sensor Aggregator. In: *Advanced Communication Technology (ICACT), 2013 15th International Conference on*, 2013 — ISBN 9788996865018, pp. 139–142.

[Dicc15] DICCIONARIO OXFORD:  
<http://www.oxforddictionaries.com/es/definicion/learner/ontology>.

[Egye07] EGYEDI, TINEKE M.: Standard-compliant, but incompatible?! In: *Computer Standards & Interfaces* vol. 29 (2007), Nr. 6, pp. 605–613.

[EnTh11] ENO, JOSHUA D and THOMPSON, CRAIG W: Virtual and Real-World Ontology Services. In: *IEEE Internet Computing* (2011), pp. 46–52.

[EPHC09] ESTEVE, MANUEL; PÉREZ-LLOPIS, ISRAEL; HERNÁNDEZ-BLANCO, LUIS E.; CLIMENTE, ALFONSO; ET AL.: SIMACET-FFT: Spanish

Army friendly force tracking system. In: *Military Communications Conference, MILCOM, 2009. IEEE, 2009* — ISBN 9781424452392, pp. 1, 7, 18–21.

[EsPP13] ESTEVE, MANUEL; PÉREZ-LLOPIS, ISRAEL; and PALAU, CARLOS E.: Friendly Force Tracking COTS solution. In: *Aerospace and Electronic Systems Magazine, IEEE* vol. 28 (2013), Nr. 1, pp. 14–21 — ISBN 0885-8985 VO - 28.

[FeGW00] FEIBUSH, ELIOT; GAGVANI, NIKHIL; and WILLIAMS, DANIEL: Visualization for situational awareness. In: *Computer Graphics and Applications, IEEE* vol. 20 (2000), Nr. 5, pp. 38–45 — ISBN 0272-1716 VO - 20.

[Goog14] GOOGLE GLASS: <http://www.google.com/glass/start/>.

[Grub95] GRUBER, T: Toward principles for the design of ontologies used for knowledge sharing. In: *International Journal of Human-Computer Studies* vol. 43 (1995), Nr. 5-6, pp. 907–928 — ISBN 10715819.

[Grup14] GRUPO DE SISTEMAS Y APLICACIONES DE TIEMPO REAL DISTRIBUIDOS (SATRD): <http://www.satrd.upv.es/>.

[HHHK10] HAN, SEUNGJU; HAN, JAE-JOON; HWANG, YOUNGKYOO; KIM, JUNG-BAE; ET AL.: Controlling virtual world by the real world devices with an MPEG-V framework. In: *Multimedia Signal Processing*

(MMSP), 2010 IEEE International Workshop on, IEEE (2010), pp. 251–256 — ISBN 9781424481125.

[HHKK13] HAN, SEUNGJU; HAN, JAE-JOON; KIM, JAMES D.K.; and KIM, CHANGYEONG: Connecting users to virtual worlds within MPEG-V standardization. In: *Signal Processing: Image Communication* vol. 28, Elsevier (2013), Nr. 2, pp. 97–113.

[HHMS11] HACID, HAKIM; HEBBAR, KARIM; MAARADJI, ABDERRAHMANE; SAIDI, MOHAMED ADEL; ET AL.: Enhancing Navigation in Virtual Worlds through Social Networks Analysis. In: *Foundations of Intelligent Systems, Lecture Notes in Computer Science*. vol. 6804 : Springer Berlin Heidelberg, 2011 — ISBN 978-3-642-21915-3, pp. 146–152.

[IbDe11] IBANEZ, JESÚS and DELGADO-MATA, CARLOS: Lessons from research on interaction with virtual environments. In: *Journal of Network and Computer Applications* vol. 34 (2011), Nr. 1, pp. 268–281.

[Inte08] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=52454](http://www.iso.org/iso/catalogue_detail.htm?csnumber=52454).

[Inte10] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=55497](http://www.iso.org/iso/catalogue_detail.htm?csnumber=55497).

[Inte13] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO/IEC 23005, Information technology -- Media context and control*.

Parts 1-7, 2013.

[JoPr10] JOVANOVA, BLAGICA and PREDÁ, MARIUS: Avatars interoperability in Virtual Worlds. In: *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*, 2010 — ISBN 9781424481125, pp. 263–268.

[Kane10] KANE, ANDREW M: *Virtual World Interoperability of Avatar Information*, Tesis doctoral, Rochester Institute of Technology, 2010.

[KCPK11] KIM, SANG-KYUN; CHUN, JONGHOON; PARK, HYUNMIN; KWON, DONGSEOP; ET AL.: Interfacing sensors and virtual world health avatar application. In: *The 5th International Conference on New Trends in Information Science and Service Science* vol. 1, lee (2011), pp. 21–25.

[KiRa13] KIPPER, GREGORY and RAMPOLLA, JOSEPH; SYNGRESS (ed.): *Augmented Reality: An Emerging Technologies Guide to AR*: Syngress, 2013 — ISBN 9781597497336.

[KJSH13] KIM, SANG-KYUN; JOO, YONG SOO; SHIN, MINHO; HAN, SEUNGJU; ET AL.: Virtual world control system using sensed information and adaptation engine. In: *Signal Processing: Image Communication* vol. 28, Elsevier (2013), Nr. 2, pp. 87–96.

[KLKR13] KIM, JAEHA; LEE, CHANG-GYU; KIM, YEONGMI; and RYU, JEHA: Construction of a haptic-enabled broadcasting system based

on the MPEG-V standard. In: *Signal Processing: Image Communication* vol. 28, Elsevier (2013), Nr. 2, pp. 151–161.

[KoMF09] KOHLER, THOMAS; MATZLER, KURT; and FÜLLER, JOHANN: Avatar-based innovation: Using virtual worlds for real-world innovation. In: *Technovation* vol. 29 (2009), Nr. 6-7, pp. 395–407.

[KSLN08] KOPENA, JOSEPH B.; SULTANIK, EVAN A.; LASS, ROBERT N.; NGUYEN, DUC N.; ET AL.: Distributed Coordination of First Responders. In: *IEEE Internet Computing* vol. 12 (2008), Nr. 1, pp. 45–47 — ISBN 1089-7801 VO - 12.

[Mert13] MERTZ, L: Preparing for the unthinkable, and taking action for when it happens. In: *Pulse, IEEE* vol. 4 (2013), Nr. 3, pp. 28–34 — ISBN 2154-2287 VO - 4.

[Meta14] METAVERSE: <http://www.metaverse1.org>.

[MGWB03] MCGRATH, SUSAN P.; GRIGG, ELIOT; WENDELKEN, SUZANNE; BLIKE, GEORGE; ET AL.: ARTEMIS: A Vision for Remote Triage and Emergency Management Information Integration (2003).

[MiCH99] MILGRAM, PAUL; COLQUHOUN, HERMAN; and HERMAN, COLQUHOUN JR: Chapter 1: A Taxonomy of Real and Virtual World Display Integration. In: *Mixed reality: Merging real and virtual worlds* (1999), Nr. JUNE 2014.

[MiCo99] MILGRAM, PAUL and COLQUHOUN, HERMAN: A taxonomy of real and virtual world display integration. In: *Mixed reality: Merging real and virtual worlds* (1999).

[PBTk05] PELLENS, BRAM; BILLE, WESLEY; DE TROYER, OLGA; and KLEINERMANN, FREDERIC: VR-WISE: A Conceptual Modeling Approach for Virtual Environments. In: *Proceedings of Virtual Concept 2005*, : Springer-Verlag, ISBN 2-287-28772-8, 2005, pp. 93–94.

[Pére09] PÉREZ-LLOPIS, ISRAEL: *Arquitectura de un sistema C4ISR para pequeñas unidades*, Tesis doctoral, Universitat Politècnica de Valencia, 2009.

[Raer15] RAE. REAL ACADEMIA ESPAÑOLA:  
[http://buscon.rae.es/drae/srv/search?id=VERUpzUOADXX2PQkV704#realidad\\_virtual](http://buscon.rae.es/drae/srv/search?id=VERUpzUOADXX2PQkV704#realidad_virtual).

[RiJo00] RICKEL, JEFF and JOHNSON, W LEWIS: Task-Oriented Collaboration with Embodied Agents in Virtual Worlds. In: *Embodied Conversational Agents, MIT Press, Cambridge, MA, 95–122* (2000).

[Saic14] SAIC: <http://www.leidos.com/products/simulation/olive>.

[Suth65] SUTHERLAND, IVAN E.: The Ultimate Display. In: *Proceedings of the IFIP Congress, 1965*, pp. 506–508.

[TGWH09] TIMMERER, CHRISTIAN; GELISSEN, JEAN; WATTL, MARKUS; and HELLWAGNER, HERMANN: Interfacing with Virtual Worlds. In: *NEM Summit*. Saint-Malo, 2009.

[VeSF11] VERDOT, VINCENT; SAIDI, ADEL; and FOURNIGAULT, LIONEL: Virtual Hybrid Communications – A Telecom Infrastructure for the Metaverse. In: *MPEG-V and Other Virtual Worlds Standards* vol. 4 (2011), Nr. 3.

[WaTR09] WANG, JIACUN; TEPFENHART, WILLIAM; and ROSCA, DANIELA: Emergency Response Workflow Resource Requirements Modeling and Analysis. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* vol. 39 (2009), Nr. 3, pp. 270–283 — ISBN 1094-6977 VO - 39.

[WTRM08] WANG, JIACUN; TEPFENHART, WILLIAM; ROSCA, DANIELA; MILEWSKI, ALLEN; ET AL.: Dynamic Workflow Modeling and Analysis in Incident Command Systems. In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* vol. 38 (2008), Nr. 5, pp. 1041–1055 — ISBN 1083-4427 VO - 38.

[YCLL10] YOON, KYOUNGRO; CHOI, BUMSUK; LEE, EUN-SEO; and LIM, TAE-BEOM: 4-D broadcasting with MPEG-V. In: *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*: IEEE, 2010 — ISBN 9781424481125, pp. 257–262.

[Yoon13] YOON, KYOUNGRO: End-to-end framework for 4-D

broadcasting based on MPEG-V standard. In: *Signal Processing: Image Communication* vol. 28, Elsevier (2013), Nr. 2, pp. 127–135.



# Capítulo 3

## Diseño de la arquitectura

### 3.1. Introducción

La tesis se enfoca en la aplicación del estándar MPEG-V a la interconexión del mundo real y mundos virtuales en un entrenador de sistemas de mando y control (C4ISR) para gestión de emergencias.

El objetivo final que se persigue es mejorar el entrenamiento de operativos en el uso de los sistemas de mando y control, lo que a su vez mejora los procedimientos de resolución de crisis. El intercambio de información entre el mundo real (MR) y el mundo virtual (MV) debe ser transparente para el usuario [Inte13], de forma que se puedan llevar a cabo operaciones de entrenamiento con un coste muy reducido y probar supuestos e hipótesis operativas que de otra manera no se podrían llevar a cabo, o resultarían difícilmente realizables en unas maniobras normales [APEP13].

Partiendo de este objetivo y considerando las hipótesis iniciales, así como las brechas en el estado del arte que se desea cubrir, se diseñó la arquitectura del sistema. A lo largo del presente capítulo se

detalla el proceso seguido. La implementación será explicada en el capítulo 4.

### **3.2. Consideraciones base**

La arquitectura a implementar debe permitir interconectar un sistema de mando y control, operando en el mundo real, con un mundo virtual. En particular, la inclusión de sensores reales en la parte virtual del sistema y el acceso a sensores virtuales desde puestos de mando desplegados en el área de operaciones. Así, los usuarios pueden entrenar en el entorno virtual con las mismas herramientas que usarían en la mitigación de una crisis real.

Para el diseño, y posterior definición de modelo de datos, se tomaron como punto de partida los siguientes datos básicos de entrada y salida de un sistema de mando y control:

- Posición de una unidad (posición + nombre unidad).
- Amenaza (posición + tipo amenaza).
- Objeto (posición + texto).
- Alarma (posición de la unidad que la genera + tipo de alarma).
- Sensor (posición + tipo sensor + valor).
- Mensajes (nombre de la unidad + texto (50 caracteres)).
- Vídeo (nombre de la unidad + URL).

Y también algunos datos que serán comunes tanto en el mundo real como en el virtual:

- Número de unidades
- Parámetros de configuración
- Hora zulu

Por otra parte tenemos aquellos que, si bien son comunes, se representarán de manera diferente en cada mundo:

- **Identidad del usuario:** En el MR: identificador, nombre de usuario, y perfil. En el MV: identificador, nombre de usuario, perfil, y preferencias.

Se puede implementar manejo de cuentas de usuario y alternancia de roles según los permisos (p. ej. un usuario al mando podría introducir modificaciones sobre el sistema conforme avanza el desarrollo de la situación).

- **Usuario – Avatar:** a toda entidad del mundo real que puede ejecutar alguna acción debería corresponder un avatar en el mundo virtual. Puede ser que dicho avatar haya sido creado previamente por un usuario como una representación de sí mismo, o que sean creados por el administrador del sistema con un perfil genérico de acuerdo al despliegue que desee crear. Incluso resulta interesante implementar la posibilidad de que el propio sistema cree entidades de manera aleatoria (ej. víctimas de un ataque).

- **Objetos:** en el MR una cadena de caracteres que incluye el identificador, descripción y valor. En el MV son lo que el estándar MPEG-V denomina objetos genéricos [Inte13] y su estructura base es muy simple: identificador, tipo, descripción, valor, acciones permitidas. Éste último campo indica la manera en que un usuario puede usar dicho objeto para interactuar con el sistema. Por ejemplo; un objeto “vehículo” permite que un usuario lo utilice como medio de transporte, mientras que un objeto “edificio” existe en el sistema pero el usuario en realidad no interactúa con él.
- **Sensores:** en el MR van por separado, en el MV son uno de los muchos tipos de objetos genéricos que pueden ser creados. Para mantener la coherencia con el despliegue de un sistema de mando y control real, se ha creado una entidad “sensor” dentro de la definición del modelo de datos MPEG-V con atributos propios que será considerada de forma independiente a otro tipo de objetos.
- **Mapas:** Actualmente en el MR los mapas se cargan a partir del fichero de misión; OLIVE utiliza mapas en formato MPF por lo que será necesario hacer traducción de un formato al otro. Esto hace parte de los desarrollos futuros.

La funcionalidad base de la pasarela es la de interconectar dos sistemas, bien sea un sistema de mando y control con un mundo virtual o dos sistemas de mundos virtuales, y hacer una traducción entre ellos. Es decir que básicamente lo que se debe garantizar es que sea capaz de tomar la información del sistema fuente, traducirla a formato MPEG-V y, posteriormente, representarla adecuadamente

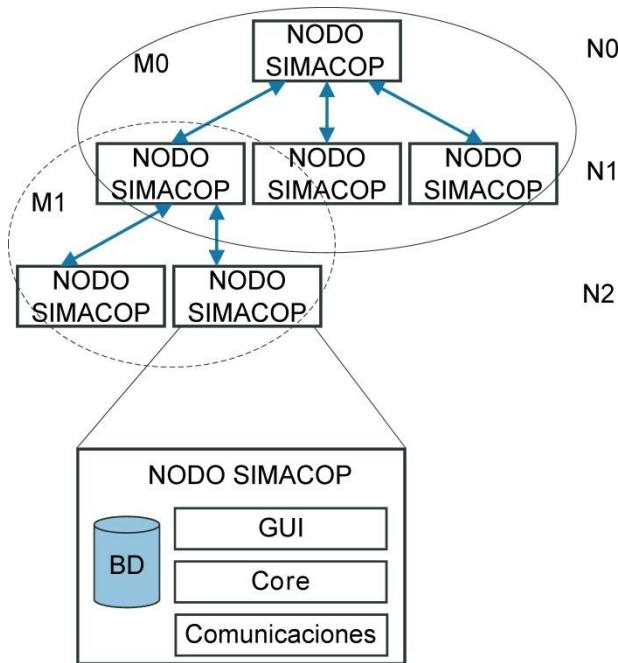
en el sistema destino. La comunicación es bidireccional.

Otra consideración importante es el despliegue de equipos en el área de entrenamiento [APEP13, APEP15]:

- Las unidades actúan como sensores móviles y actuadores en la mitigación de la crisis simulada, alimentando el sistema con datos reales y respondiendo a las órdenes de las personas al mando.
- En el mundo real estas unidades están conectadas al centro operativo de mando mediante algún tipo de sistema de comunicaciones, tal como una red MESH, WiMax o un terminal satelital.
- Los sensores desplegados en el mundo real serán accesibles desde el mundo virtual y viceversa: la posición e identificador de las unidades creadas y distribuidas en el mundo real serán accesibles desde terminales del sistema de mando y control implementado en el mundo virtual.

Vale la pena resaltar que el sistema debe ajustarse a las arquitecturas de los sistemas de mando y control, así como de mundos virtuales. Si bien se ha partido de dos aplicaciones específicas, SIMACOP y OLIVE, se pretende ofrecer una arquitectura modular y escalable que sea fácilmente modificada para integrar nuevas aplicaciones.

Fue necesario analizar las arquitecturas usadas comúnmente en sistemas C4ISR y mundos virtuales [Thom11]. Parte de esto ya fue comentado previamente, en el capítulo 2.



**Figura 15. Arquitectura de un C4ISR.**

Los sistemas de mando y control usualmente tienen una arquitectura que refleja dos dimensiones principales: la estructura organizacional, con una división jerárquica ( $N0, N1, N2 \dots$ ); y la estructura de red, compuesta por las distintas mallas de red ( $M0, M1 \dots$ ). La arquitectura es distribuida, con consistencia basada en un mecanismo de réplica y sincronización no estricta (recordemos que usualmente en comunicaciones en entornos tácticos, como los datos tienen un periodo de validez determinado, es posible, e incluso aceptable, que se pierdan algunos paquetes de información). Ver Figura 15.

Por otro lado, para sistemas de mundo virtual (y considerando las particularidades de OLIVE), nos basaremos en una arquitectura con servidores asociados con base de datos centralizada, sincronización

estricta (esto quiere decir que toda información crítica debe ser actualizada dentro del mismo intervalo de tiempo) y consistencia rigurosa, que determina la granularidad temporal y de eventos. Se muestra en la Figura 16.

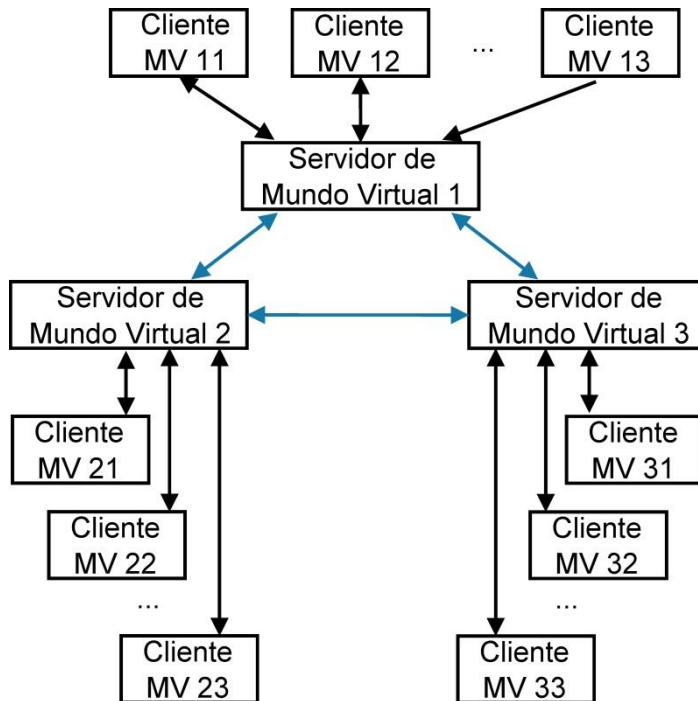


Figura 16. Arquitectura MV con servidores asociados.

Con esta arquitectura se mejora el balance de carga y la escalabilidad con respecto al número de usuarios y uso de recursos. Además, en el caso de que algún componente falle, no colapsaría todo el sistema; por lo que ofrece mayor seguridad ante ataques de seguridad.

Para garantizar que el sistema implementado tenga validez desde el punto de vista de las funciones buscadas (entrenamiento táctico,

consistencia de los datos, etc.), fue necesario resolver el mapeo temporal y semántico de entidades entre el mundo real y mundo virtual.

El *mapeo semántico* determina cómo ajustar las entidades, su taxonomía, sus funcionalidades y las interrelaciones de un mundo en las entidades del otro. En la Figura 17, y a modo de ejemplo, se muestra el mapeo de entidades con una granularidad gruesa.

Mundo Real		Mundo Virtual	
Unidad	↔	Avatar (o grupo de avatares)	
Amenaza	↔	Símbolo de amenaza	
Objeto	↔	Marca espacial	
Flujo de vídeo	↔	Sensor	

Figura 17. Mapeo semántico a alto nivel.

Por otra parte, tenemos el *mapeo temporal*. En un mundo virtual la temporización y sincronización entre nodos es estricta, en cambio en el mundo real el acople temporal es más relajado puesto que los requisitos de tiempo real en entornos de comunicaciones adversos hacen inviable la sincronización estricta pero esto, como ya hemos comentado anteriormente, es aceptable en ciertos momentos.

Para adaptar las dinámicas temporales de ambos mundos y solventar posibles incoherencias, en el laboratorio se han debido adaptar mecanismos que eviten las incoherencias que los datos del MR podrían introducir en el MV y permitir que la mayor rigidez del MV coexista con el MR. Además, se han adecuado las tasas y dinámicas temporales a una mayor velocidad de actualización de la información



del mundo virtual. Cabe señalar que ésta es una línea de investigación no cerrada y que probablemente se continúe dentro del grupo en los próximos tiempos.

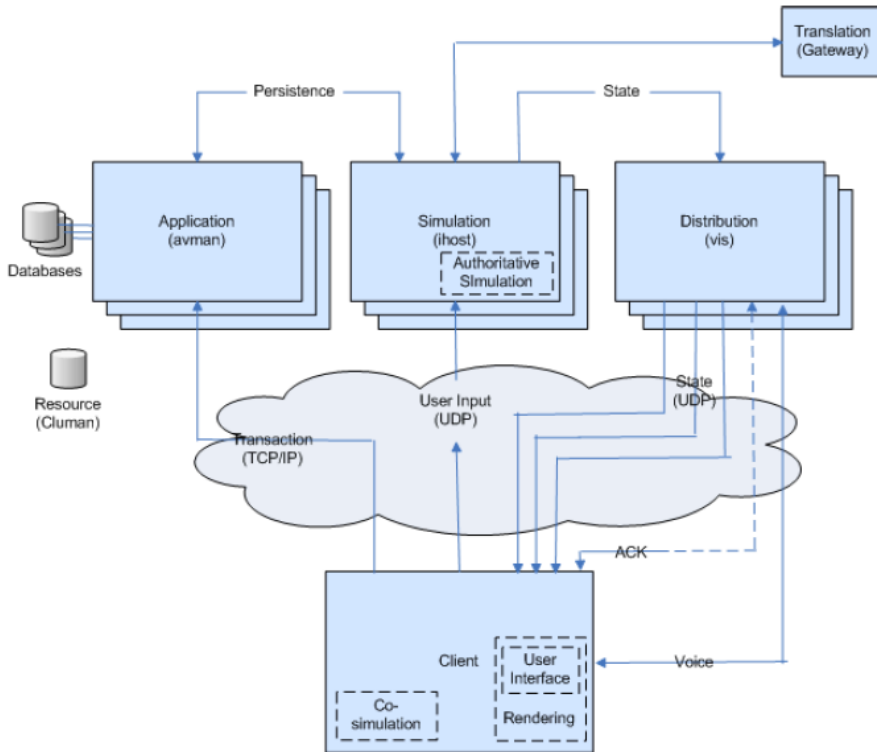
### **3.2.1. Generación de mundos virtuales**

Existen muchas herramientas, tanto propietarias como de código abierto, para generar mundos virtuales. Para el desarrollo de la presente tesis doctoral, como se comentó anteriormente en la sección 2.3, se ha utilizado la plataforma propietaria de software OLIVE [Saic14].

OLIVE, *On-Line Interactive Virtual Environment*, es una plataforma de software propietaria de la empresa estadounidense SAIC que proporciona capacidades de comunicación interactiva multimedia para colaboración, entrenamiento, operaciones y educación. Permite que clientes, socios y desarrolladores creen mundos virtuales persistentes en los cuales los usuarios pueden colaborar a través de redes de comunicaciones [CFPL08] para conseguir un fin común. Los usuarios finales se pueden conectar bien sea al clúster público de la compañía, por ejemplo para solicitar soporte, o a algún mundo virtual que haya sido creado y configurado en un servidor propio.

La plataforma está diseñada para proporcionar una experiencia de usuario distribuida, colaborativa y enriquecida en un entorno virtual 3D similar al mundo real. Es en sí misma un conjunto de aplicaciones, herramientas e interfaces que permiten crear contenidos y escenarios específicos, además de crear código propietario para extender las funcionalidades y desarrollar

aplicaciones de mundo virtual a medida. Esto último se hace a partir del OLIVE SDK (*Software Development Kit*, kit de desarrollo de software).



**Figura 18. Principales componentes de la arquitectura de OLIVE.**

OLIVE está basada en una arquitectura cliente/servidor en la cual los clientes se conectan a un servidor central a través de una LAN, WAN, o Internet. Asimismo, separa las capacidades en módulos cliente y servidor. De esta forma, es posible configurar desde aplicaciones monousuario en una sola ubicación física, hasta aplicaciones a gran escala soportando cientos de usuarios concurrentes y que pueden estar distribuidos geográficamente

alrededor del mundo. Los principales componentes de la arquitectura de OLIVE se muestran en la Figura 18 [Saic10].

Durante el desarrollo del proyecto se han configurado múltiples instancias de clientes de OLIVE sobre Microsoft Windows<sup>®</sup> XP y Microsoft Windows<sup>®</sup> 7, así como de dos servidores con diferentes sistemas operativos: el primero Windows Server 2008 R2 y el segundo CentOS 5.0.

Los clientes del mundo virtual son instanciados simulando el teatro de operaciones del mundo real (a partir del fichero de misión) y se conectarán a la pasarela de interconexión (ver sección 3.4) donde se hace la traducción de datos a MPEG-V para su posterior envío hacia el mundo real, o hacia otros mundos virtuales si es el caso. De manera similar, la pasarela se conectará con el mundo real, por lo que también habrá que hacer un proceso de detección y representación de las señales.

Una anotación importante es que como OLIVE no fue diseñada bajo los lineamientos del estándar MPEG-V (cabe señalar que actualmente no existe ningún producto en el mercado para generar mundos virtuales que cumpla dicha normativa), ha sido necesario desarrollar el modelo de datos equivalente para OLIVE en formato MPEG-V, para poder enviar y recibir datos normalizados desde y hacia la pasarela de interconexión. Sucede lo mismo con SIMACOP, y también se ha definido el modelo de datos equivalente para resolver el intercambio de datos. Profundizaremos más al respecto en la sección 4.2.

### 3.3. El entrenador táctico

A partir de todo lo comentado hasta ahora, se definió la arquitectura del sistema con tres componentes principales (ver Figura 19).

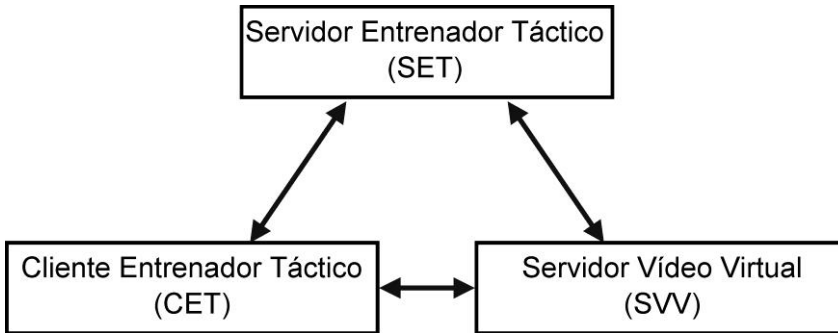


Figura 19. Componentes principales del sistema.

Veamos cada componente con mayor detalle:

- **Servidor Entrenador Táctico (SET):** donde se lleva a cabo la configuración del sistema. Incluye el servidor de mundos virtuales y es donde se ejecuta la pasarela de interconexión, el elemento clave para conseguir interoperabilidad y que se explicará detalladamente en la sección 3.4).
- **Cliente Entrenador Táctico (CET):** es el subsistema cliente a que utilizan los nodos de entrenamiento; incluye el sistema de mando y control del mundo real y un cliente de mundo virtual.
- **Servidor de Vídeo Virtual (SVV):** un innovador sistema para la gestión, distribución y reproducción de los flujos de vídeo generados por los sensores en el campo de operaciones, de fuentes tanto reales como virtuales. Su arquitectura se muestra en

la Figura 20.

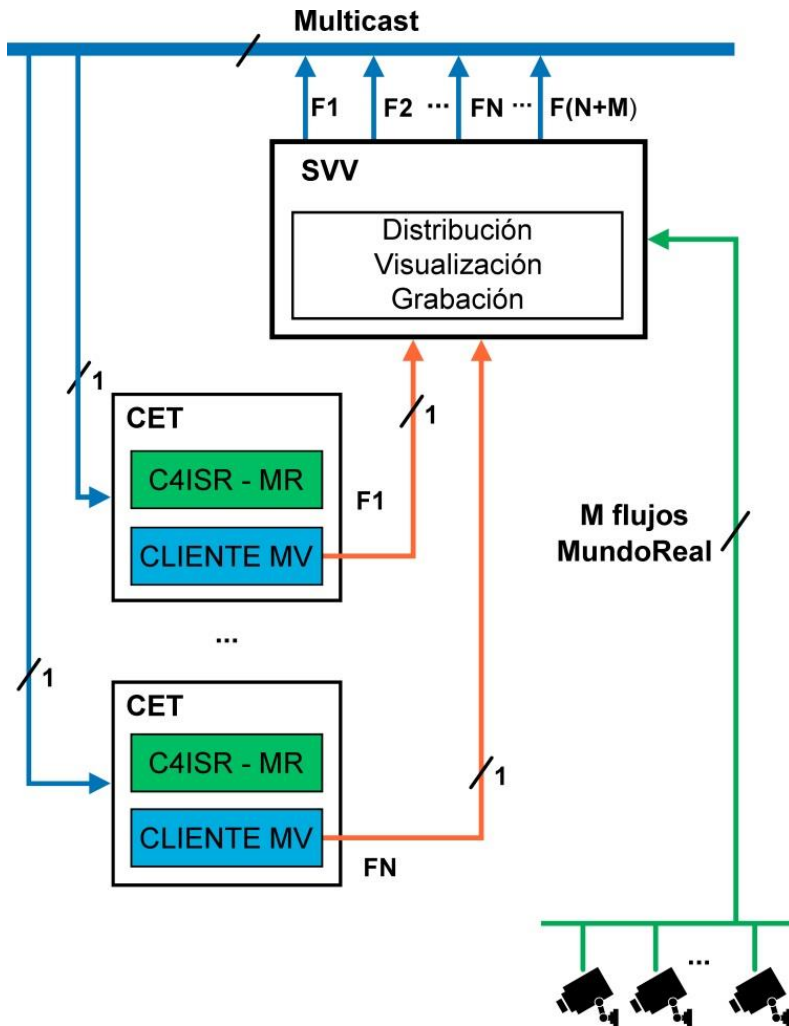


Figura 20. Arquitectura de video.

La Figura 21 ilustra un despliegue típico de los componentes del entrenador táctico. Como se puede observar, tenemos un conjunto de clientes entrenador táctico (CET) junto con un nodo central donde está el servidor entrenador táctico (SET).

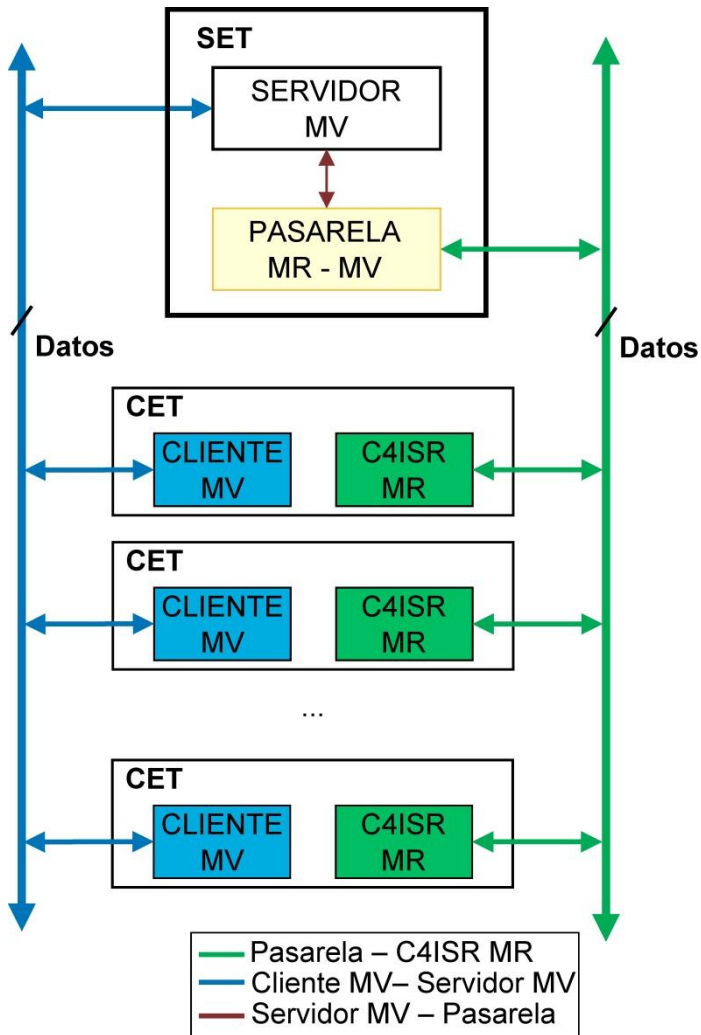


Figura 21. Arquitectura del sistema.

La información generada en el C4ISR MR (sistema de mando y control del mundo real) va directamente a la pasarela MR – MV, la cual la transforma en información relevante para el MV y se la transmite al servidor de MV. Por otra parte, y de manera similar, la información que estimamos oportuna del MV es recibida por la

pasarela y enviada a los nodos (a cada uno la que procede) del MR.

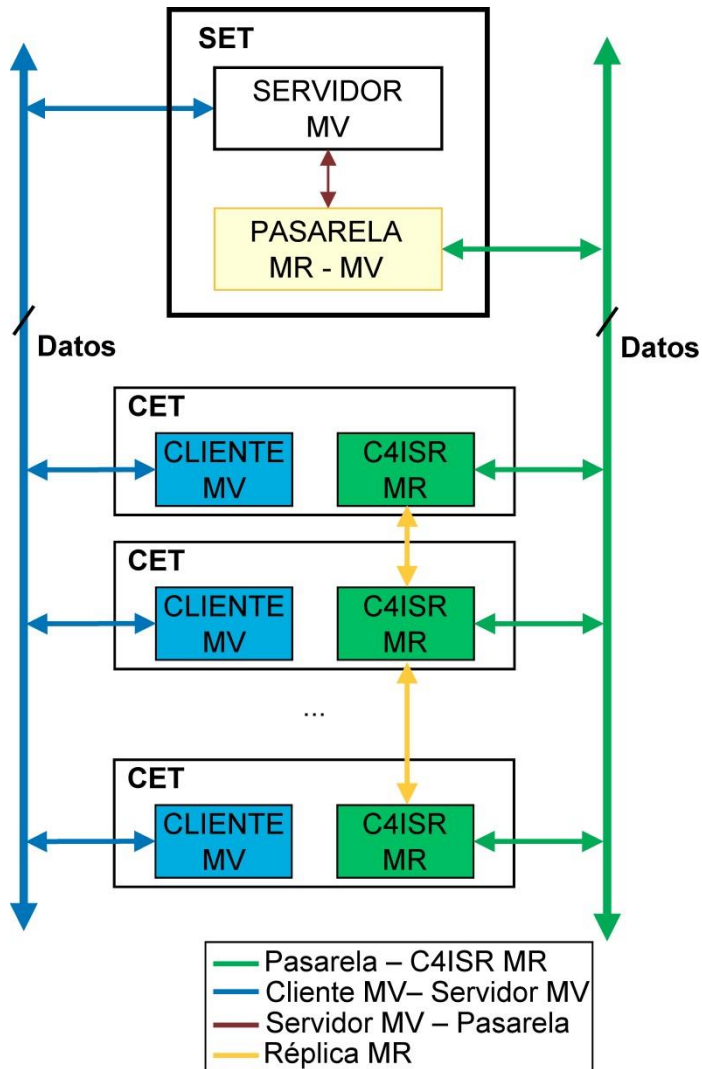


Figura 22. Arquitectura del sistema ortogonal a la réplica del MR.

Una directriz de diseño que se ha seguido en todo el desarrollo es implementar el sistema de forma que no interfiera con las condiciones normales de funcionamiento de los sistemas ya

existentes. Así, al esquema de réplica distribuida de la información entre los nodos del MR le es completamente transparente la interconexión con los mundos virtuales. En la Figura 22 se puede ver el añadido de esa réplica distribuida, siguiendo la estructura descrita anteriormente en la Figura 15.

Es por ello que la ruta de datos en la pasarela de/hacia los nodos del MR sigue el esquema mostrado en la Figura 23:

MR → MV: del CETi – MR al SET  
MV → MR: del SET al CETi – MR que sea productor de dicha información en el mundo real.

**Figura 23. Ruta de los datos entre nodos.**

Los flujos de información que se transportan entre MR y MV son:

- Información relativa a la naturaleza y localización de unidades/avatares.
- Amenazas y objetos.
- Flujos de vídeo.

Como ya se ha comentado, una contribución novedosa y destacada del trabajo realizado es la utilización del estándar MPEG-V para la interconexión [Inte13]. Se ha desarrollado un modelo de datos basado en MPEG-V (ver sección 4.2) que permite llevar a cabo el mapeo de información entre MR y MV, y un paradigma de comunicación basado en servicios web con un esquema petición-respuesta del cual hablaremos en el numeral 3.5.



En la Figura 24 se puede ver la arquitectura en dos capas de la solución MPEG-V desarrollada. La capa superior implementa un modelo de datos MPEG-V genérico mientras que en la inferior se lleva a cabo una adaptación particular para cada sistema, ya sea del mundo real o mundo virtual, y considerando las peculiaridades de los modelos de datos utilizables en un escenario de gestión de emergencias.

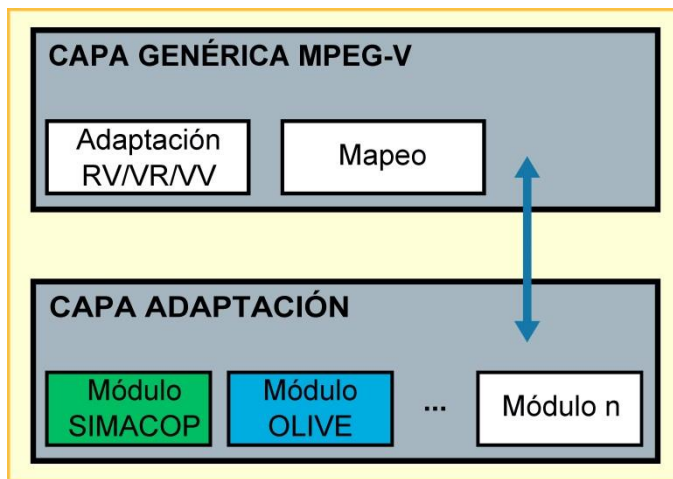


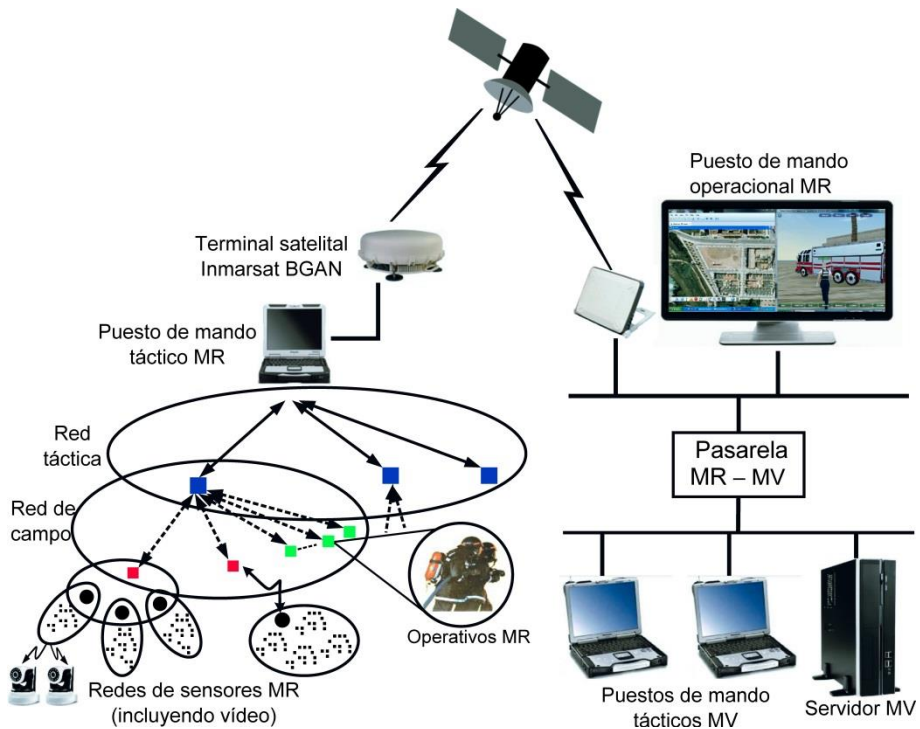
Figura 24. Arquitectura en dos capas para la implementación de MPEG-V.

### 3.4. La pasarela de interconexión

Para conseguir interoperabilidad entre el mundo real y mundos virtuales, se ha desarrollado una **pasarela de interconexión** basada en los formatos de datos definidos en el estándar MPEG-V. Su principal funcionalidad es interconectar dos sistemas (ya sea un mundo virtual con un sistema de mando y control o varios mundos virtuales entre sí), traducir y adaptar la información relevante de

acuerdo al formato de datos MPEG-V y mapearla entre ellos. Este es uno de los principales aportes de la tesis doctoral.

La ubicación de la pasarela en el sistema de entrenamiento en C4ISR para gestión de emergencias se muestra en la Figura 25.

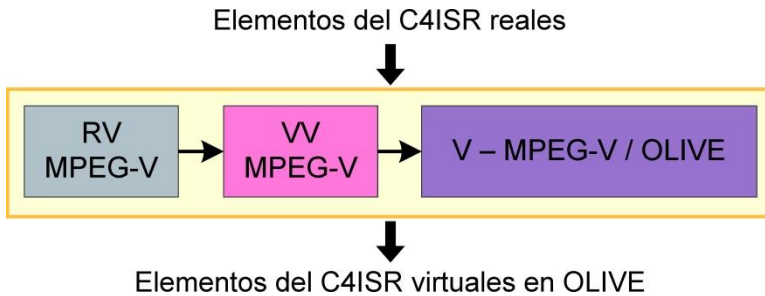


**Figura 25. Ubicación de la pasarela de interconexión MR - MV.**

La pasarela de interconexión debe ser capaz de: (i) tomar los datos del sistema de mando y control, (ii) convertirlos a datos de mundo virtual en formato MPEG-V y (iii) adaptarlos adecuadamente para entregarlos al mundo virtual de destino. La comunicación es bidireccional.

La Figura 26 muestra de manera conceptual el funcionamiento de la

pasarela. El flujo de datos se puede describir así: los datos originados en el sistema C2IS real (SIMACOP) son traducidos a formato MPEG-V. A continuación, deben ser traducidos al formato que maneja OLIVE para, finalmente, entregarlos a la aplicación y poder representarlos adecuadamente.



**Figura 26. Esquema conceptual de la pasarela de interconexión.**

El camino inverso sería similar: tomar los datos generados del mundo virtual, enviarlos a la pasarela para convertirlos a formato MPEG-V, traducirlos al formato del sistema de mando y control, y entregarlos para su representación. La diferencia radica en que en el primer paso el flujo de información no será real → virtual sino virtual → virtual.

Para garantizar la escalabilidad del sistema y facilitar la posterior incorporación de nuevas herramientas, se ha optado por desarrollar una arquitectura modular (ver Figura 27) implementando una capa genérica donde se ejecutan las funciones de adaptación y traducción de datos siguiendo la normativa MPEG-V. De esta manera, se hace posible conectar cualquier tipo de sistema C4ISR con cualquier mundo virtual que implemente el sistema de entrenamiento en gestión de emergencias.

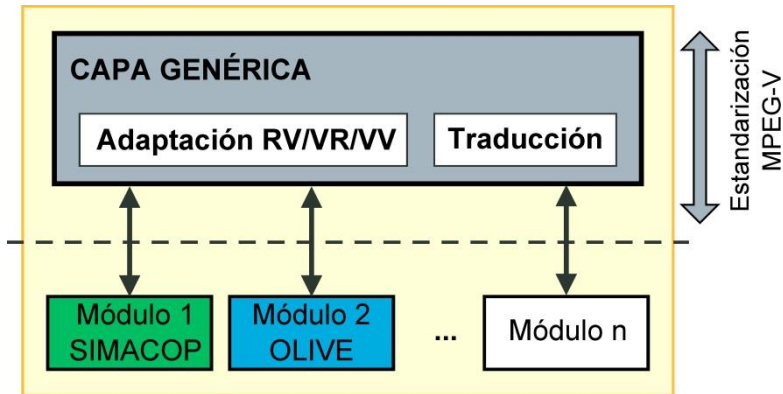


Figura 27. Arquitectura modular para la interconexión de sistemas.

Con todo esto, se propuso la arquitectura de la pasarela de interconexión (Figura 28) [Ardi13]. Se pueden ver más claramente los elementos que la componen y cómo se conectan entre sí.

### 3.5. Mecanismo de comunicación

La Figura 29 ilustra el paradigma de comunicación a implementar: cuando un cliente (1) hace una petición (actualiza su información asociada) (2), bien sea desde mundo virtual o real, ésta se encapsula y envía como XML, mediante un servicio web y vía TCP/IP, entregándolo (3) a la capa de adaptación y traducción con el fin de realizar la adaptación RV, VR o VV según corresponda y de convertirlo a formato MPEG-V siguiendo el modelo de datos definido (la implementación del modelo de datos se detalla en la sección 4.2). Los datos resultantes se encapsulan de nuevo en un documento XML que, mediante el mismo servicio web, (4) será entregado al módulo que corresponda (5) para ser representados (6), y queda a la espera de novedades (que ocurrirían en cuanto algún otro cliente

notifique cambios en su información asociada).

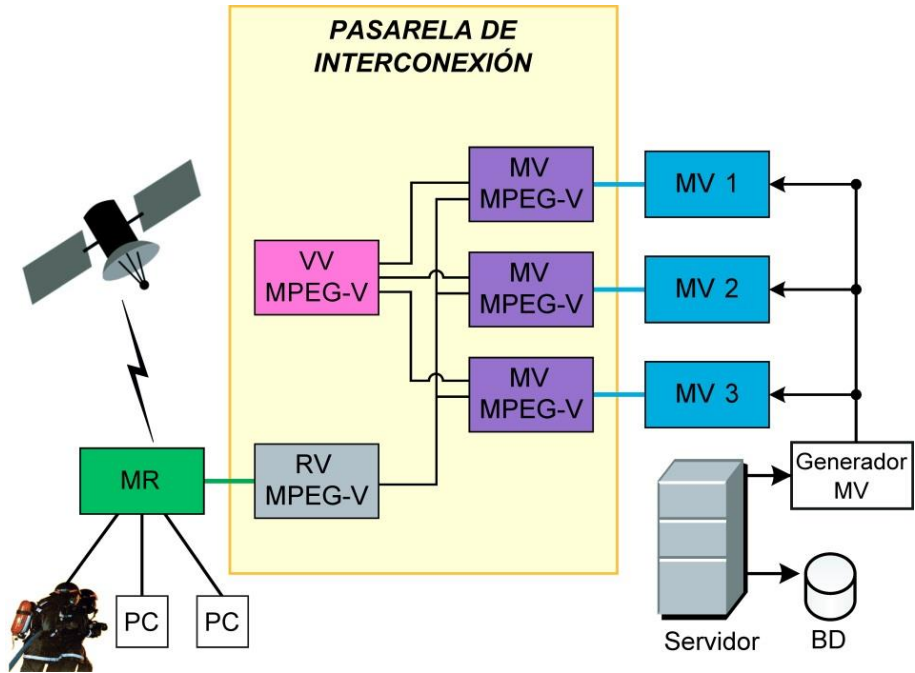


Figura 28. Arquitectura de la pasarela de interconexión.

La respuesta viajará en el camino inverso. El servicio web entrega la respuesta a la capa de adaptación/traducción, donde se efectuarán las acciones pertinentes y, acto seguido, se transferirá (vía TCP/IP) el resultado al cliente que hizo la petición.

De esta forma, cualquier sistema de mando y control del mundo real podría utilizar toda la arquitectura desarrollada si implementase un módulo de mapeo de datos y comunicación que se adhiriese a las especificaciones del modelo que se está proponiendo en la presente tesis.

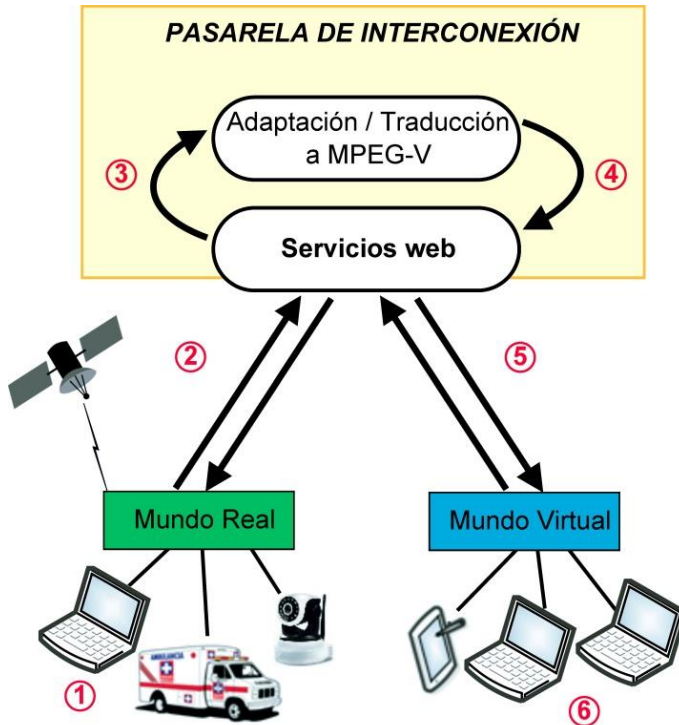


Figura 29. Mecanismo de comunicación

Como puede intuirse, habrá que vigilar que la información tanto en el mundo real como en el virtual se mantenga actualizada y que los datos sean válidos en todo momento, para garantizar que los XML generados sean correctos. También, en etapas futuras, se podrían definir parámetros de control relacionados con la gestión adecuada de un sistema de tiempo real.

### 3.6. Bibliografía

[APEP13] ARDILA, LAURA; PÉREZ-LLOPIS, ISRAEL; ESTEVE, MANUEL; y PALAU, CARLOS E.: LVC Training Environment for Strategic and

Tactical Emergency Operations. En: *10th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2013)*. Baden-Baden, Germany, 2013 — ISBN 9783923704804, pp. 140-144.

[APEP15] ARDILA, LAURA; PÉREZ-LLOPIS, ISRAEL; ESTEVE, MANUEL; y PALAU, CARLOS E.: Interoperable architecture for joint real/virtual training in emergency management using the MPEG-V standard. En: *Computer Standards & Interfaces* vol. 41 (2015), pp. 39-55.

[Ardi13] ARDILA, LAURA: *Pasarela de interconexión entre sistemas C4ISR y mundos virtuales aplicando el estándar MPEG-V*, Tesina de Máster, Máster Universitario en Tecnologías, Sistemas y Redes de Comunicaciones, Universitat Politècnica de València, 2013.

[CFPL08] CHEN, YUNG-FANG; FREITAS, SARA DE; PARKER, EL; LIAROKAPIS, FOTIS; ET AL.: The use of virtual world platforms for supporting an emergency response training exercise. En: *CGAMES conference*, 2008.

[Inte13] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO/IEC 23005, Information technology -- Media context and control, parts 1-7*, 2013.

[Saic10] SAIC: OLIVE SDK - Guía para desarrolladores (2010).

[Saic14] SAIC: <http://www.leidos.com/products/simulation/olive>.

[Thom11] THOMPSON, CRAIG W: Virtual World Architectures. En: *IEEE Internet Computing* vol. 15, leee (2011), Nr. 5, pp. 11-14.



# Capítulo 4

## Implementación

### 4.1. Introducción

Una vez especificados los antecedentes y requerimientos del sistema, así como el diseño de la arquitectura, entramos a detallar el proceso de implementación.

El modelo de datos (numeral 4.2), definido a partir del estudio exhaustivo del estándar MPEG-V y de los requisitos del sistema, es indispensable para hacer la traducción y adaptación de datos en la pasarela de interconexión (explicada anteriormente, en la sección 3.4). La implementación de cada componente del entrenador táctico se puede consultar en el numeral 4.3. Por último, en la sección 4.4, se explica la implementación del paradigma de comunicación, cuya arquitectura fue descrita en la sección 3.5.

### 4.2. Modelo de datos

Teniendo en cuenta que uno de los objetivos de la tesis es ofrecer una solución que implemente formatos de datos estandarizados

según la normativa MPEG-V, fue necesario hacer una abstracción de la sintaxis y semántica de los datos que se deben transmitir entre los nodos del sistema entrenador táctico para la adecuada ejecución de ejercicios de entrenamiento en gestión de emergencias.

A partir de dicha abstracción, y con el objetivo de facilitar la posterior integración de aplicaciones, se definió un modelo de datos genérico para el entrenador táctico aplicando la normativa MPEG-V (ver sección 4.2.1). Éste constituye el modelo base del sistema y será extendido para ajustarlo a las particularidades de cada herramienta que se quiera integrar.

El modelo de datos ha sido implementado en la pasarela de interconexión constituyendo una capa adicional e independiente al sistema donde, además, se lleva a cabo la traducción entre formatos y la adaptación RV (real a virtual), VR (virtual a real), o VV (virtual a virtual); según corresponda, y se envían los datos hacia los demás módulos.

Los correspondientes modelos extendidos y adaptados para SIMACOP y OLIVE se resumen en las secciones 4.2.2 y 4.2.3. La definición de datos completa puede ser consultada en el Anexo B.

Vale la pena señalar que todos los modelos de datos propuestos en esta tesis además de cumplir la normativa MPEG-V, cumplen también las recomendaciones de la W3C [Wor15].

#### **4.2.1. Modelo de datos MPEG-V genérico**

Consta de cuatro elementos base (ver Figura 30):

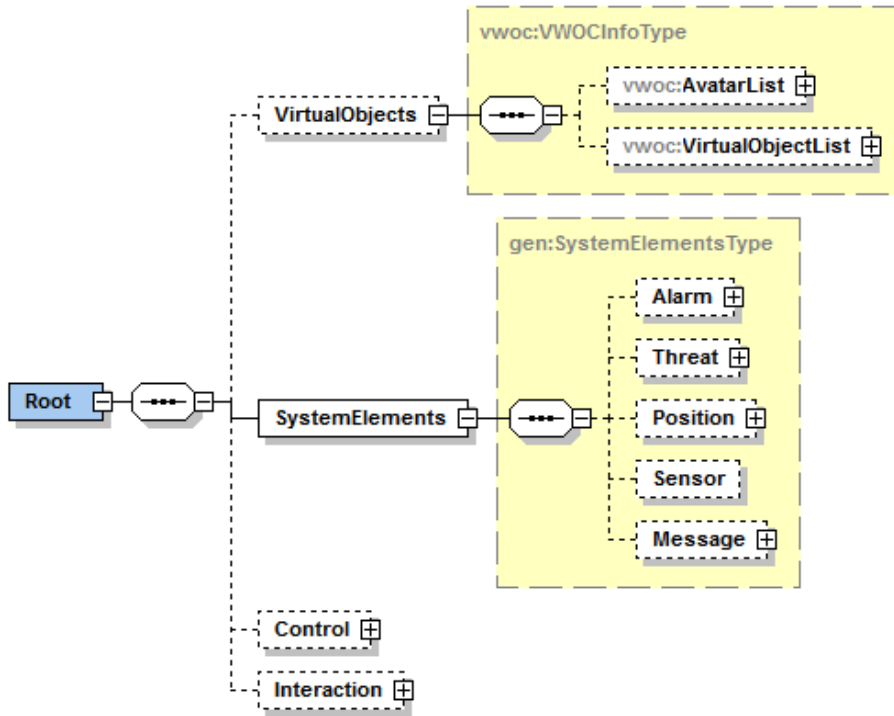


Figura 30. Definición del formato de datos genérico.

- *VirtualObjects*: define la información relativa a avatares y objetos de mundo virtual aplicando la normativa de MPEG-V parte 4 y considerando lo comentado en la sección 2.4.
- *SystemElements*: describe los elementos genéricos del sistema C4ISR que se intercambian entre nodos del entrenador táctico: alarma, amenaza, posición, sensor y mensaje. La definición de estos elementos se muestra en las Figuras 31 a 35.
- *Control*: descripción de la información de control a partir del formato de datos definido en la parte 2 de MPEG-V.

- *Interacción*: comandos para control de dispositivos y descripción de la información adquirida a través de sensores. Extiende el formato de datos de MPEG-V parte 5.

En futuras ampliaciones del sistema, podrían incluirse las definiciones de elementos tipo *SensedInfoBaseType* y *DeviceCommandBaseType* para especificar las capacidades de tipos individuales de sensor y comandos específicos para cada tipo de sensor, respectivamente.

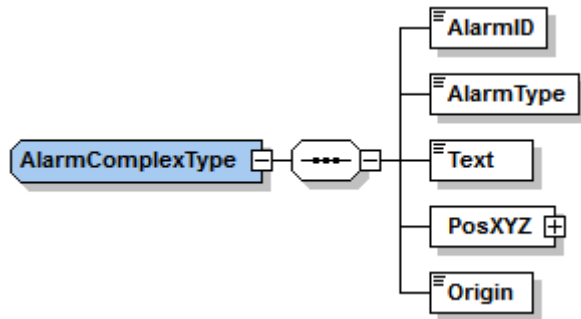


Figura 31. Estructura elemento *Alarm* (alarma).

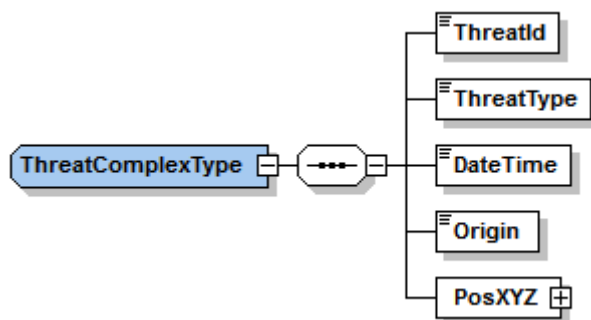


Figura 32. Estructura elemento *Threat* (amenaza).

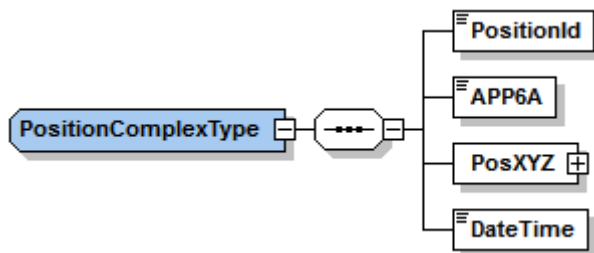


Figura 33. Estructura elemento *Position* (posición).

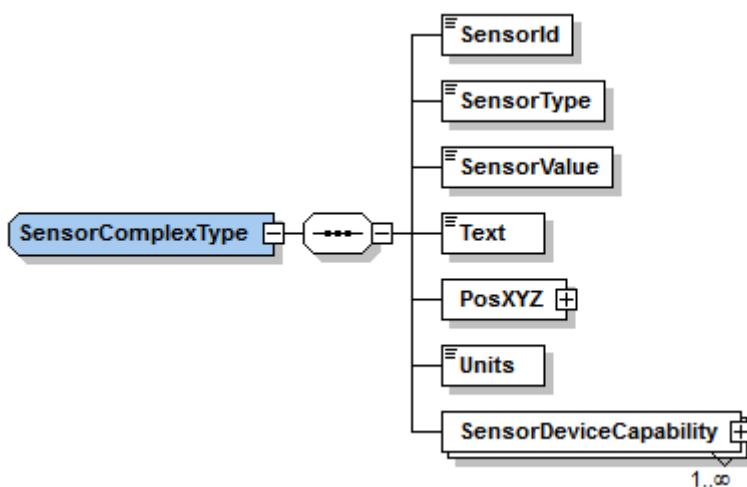


Figura 34. Estructura elemento *Sensor* (sensor).

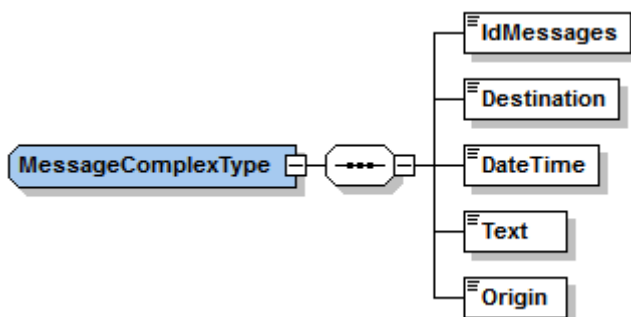


Figura 35. Estructura elemento *Message* (mensaje).

#### 4.2.2. Modelo de datos MPEG-V para SIMACOP

A partir del análisis de la sintaxis y semántica del C4ISR, y de la abstracción de qué tipo de información es necesario transmitir a otros módulos para garantizar la consistencia de datos y una adecuada implementación del entrenador táctico, se propuso la definición de datos para SIMACOP en formato MPEG-V.

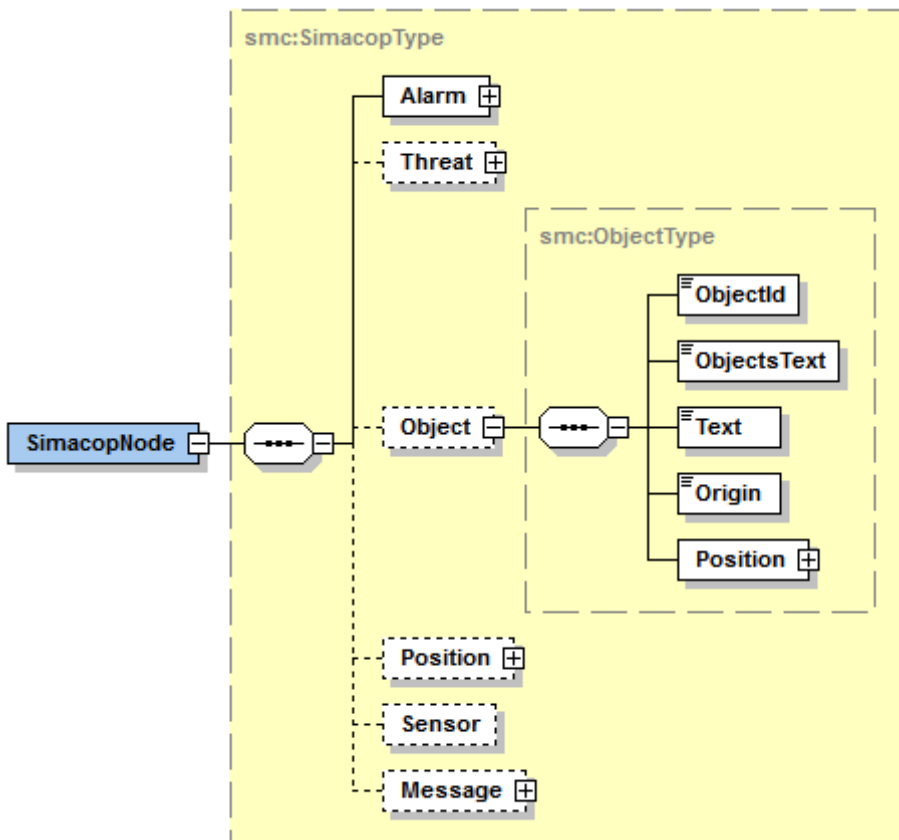


Figura 36. Definición formato de datos equivalente en MPEG-V para SIMACOP.

Como es de esperar, los elementos definidos en el esquema genérico están presentes, y se introduce además el elemento propio

de SIMACOP para representar un objeto, *Object*.

### 4.2.3. Modelo de datos MPEG-V para OLIVE

Siguiendo el mismo orden de ideas, se analizó con detalle la herramienta para generación de mundos virtuales, OLIVE. Se determinó qué elementos sería necesario describir en el esquema XML y cuáles debían extender del formato MPEG-V. Como resultado, se definió el esquema de representación de datos equivalente. Se muestra en la Figura 37.

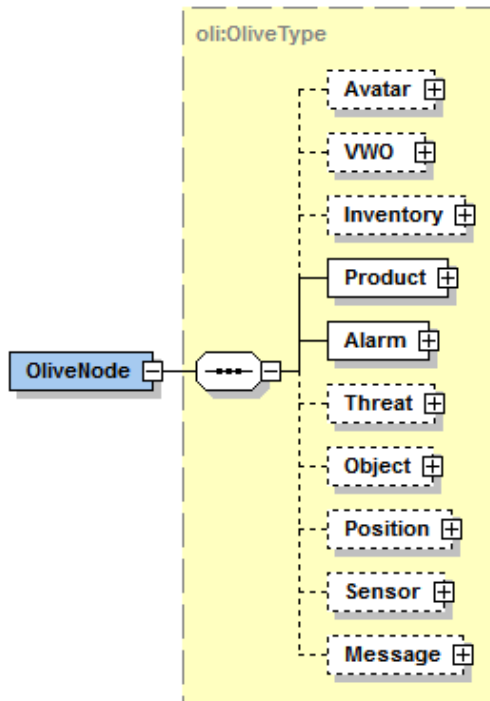


Figura 37. Definición formato de datos equivalente en MPEG-V para OLIVE.

Aparte de estar presentes los elementos básicos del sistema C4ISR que serán mapeados del mundo real al virtual, extendiendo así el

modelo genérico del sistema, se han definido algunos elementos adicionales para abstraer información de representación propia de OLIVE: avatar (*Avatar*), producto (*Product*), inventario (*Inventory*), y objeto de mundo virtual (*VWO*). Dichas definiciones son ilustradas en las Figuras 38 a 41.

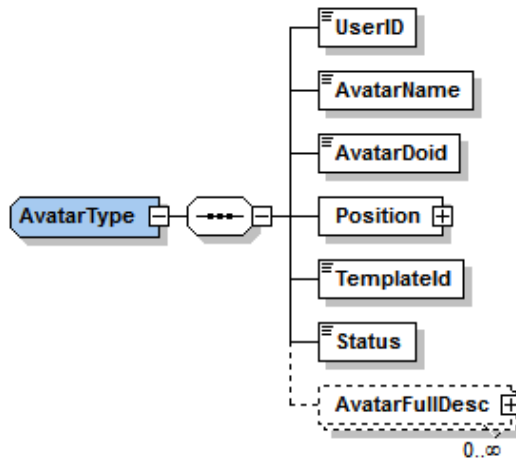


Figura 38. Estructura elemento *Avatar* (avatar).

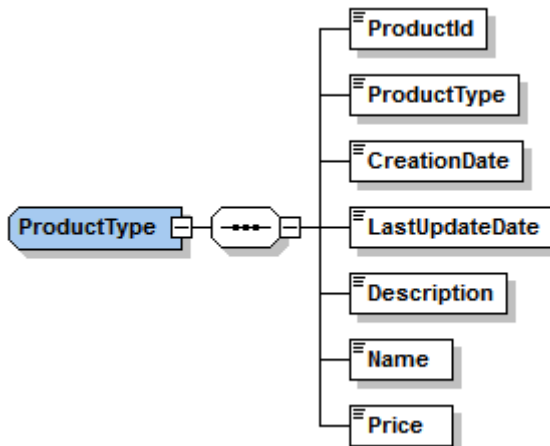


Figura 39. Estructura elemento *Product* (producto).



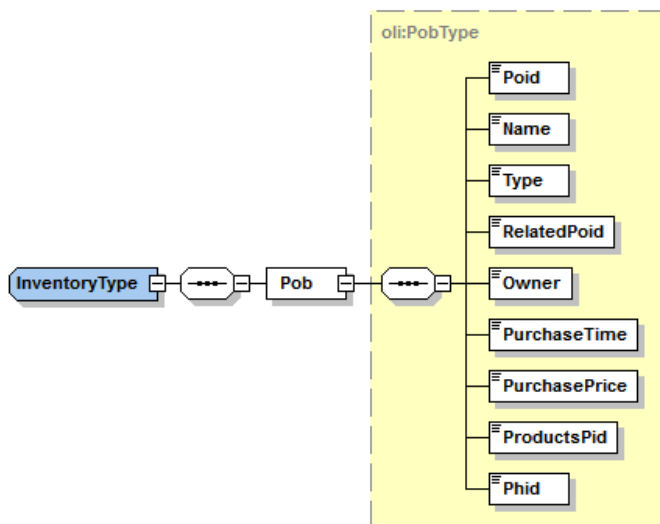


Figura 40. Estructura elemento *Inventory* (inventario).

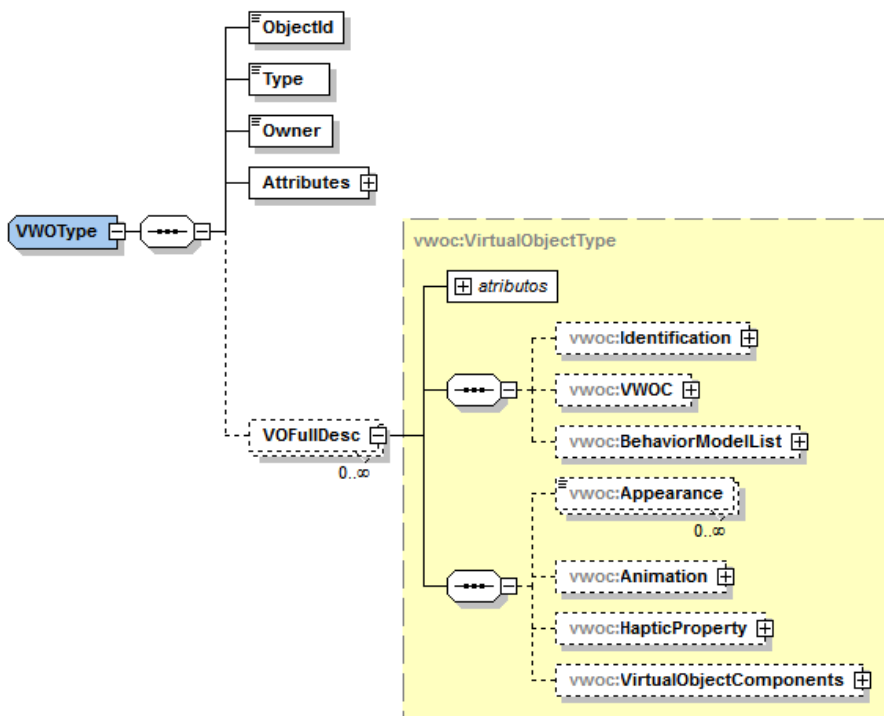


Figura 41. Estructura elemento *VWO* (objeto de mundo virtual).

Una característica común con SIMACOP es la presencia del elemento *Object* (objeto), si bien representan elementos distintos del sistema (y, a su vez, ésta es una entidad diferente de la definida con *VWO*, que es de uso interno de OLIVE) con lo que deben ser definidos diferenciadamente para cada herramienta. Así, la entidad *Object* de OLIVE incluye también elementos de representación de entidades normalizados por el estándar MPEG-V. Su definición se muestra en la Figura 42.

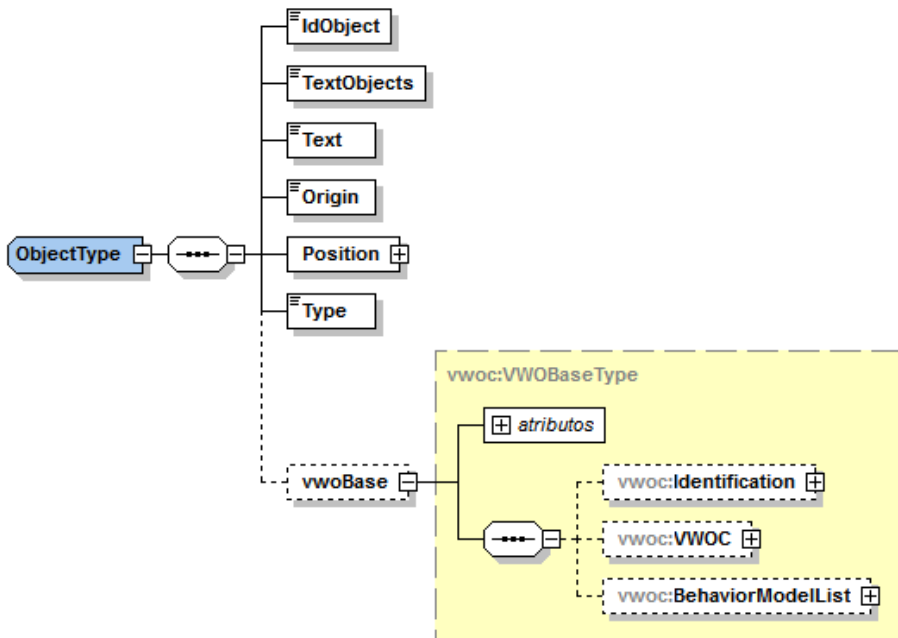


Figura 42. Definición elemento *Object* (objeto) para OLIVE.

### 4.3. El entrenador táctico

La principal característica y caso de uso del entrenador táctico (en adelante, ET) es proporcionar una aplicación que permite integrar

dentro del sistema de mando y control SIMACOP tanto elementos del mundo real como elementos del mundo virtual. En el contexto de la presente tesis doctoral, ha sido utilizado también para validar la pasarela MPEG-V.

La arquitectura del ET fue explicada anteriormente, en la sección 3.3. En una implementación típica se tendrá un equipo servidor SET que incluye el servidor de mundos virtuales (SMV) y la pasarela de interconexión; un conjunto indeterminado de nodos cliente CET (cada uno de los cuales incluye un sistema de mando y control del MR y un cliente de MV) y un SVV para la gestión de vídeo.

El sistema es escalable, admitiendo cientos de usuarios y objetos reales y/o virtuales que pueden interactuar entre sí, pero ahorrando los costes asociados que implicaría llevar a cabo el mismo ejercicio en un escenario real.

#### **4.3.1. Servidor Entrenador Táctico**

Como ya se comentó, es donde se lleva a cabo la configuración del sistema y se ejecuta la pasarela que interconecta el mundo real con uno o varios mundos virtuales.

El elemento fundamental del SET es un proceso que está continuamente a la escucha de novedades y que, al recibir las, mapea consecuentemente la información de un mundo a otro a través de la pasarela de interconexión. Recordemos que es en la pasarela donde los datos son adaptados y traducidos siguiendo el formato de datos propuesto en la sección 4.2.

### **Configuración global**

El sistema necesita de una estructura organizativa de las unidades implicadas en una misión (bomberos, policías, militares, etc.), así como la estructura de red de las distintas mallas de comunicaciones implicadas. Para ello, en SIMACOP y otros sistemas de mando y control similares, se utilizan los denominados ficheros de misión (FDM) que reflejan toda esa información. Se ha decidido mantener dicha aproximación de FDM para ser completamente compatibles a nivel de configuración, ampliando la estructura para reflejar los nuevos elementos configurables de la interconexión con el mundo virtual.

Entonces, en toda operación del ET tendremos:

- FDM del sistema de mando y control del mundo real (MC-MR).
- FDM de los clientes CET que apunta al FDM MC-MR.
- FDM de la estructura de interconexión, sólo visible en el SET.
- FDM de cámaras, visible únicamente en el SVV.

De esta forma, se utiliza un FDM del MC-MR ya existente, y el administrador del sistema crea un FDM para el SVV y otro para los distintos clientes CET. Este proceso es transparente para los usuarios del sistema, y lo único que deberán hacer es cargar los ficheros que corresponden desde la interfaz de usuario.

Por otra parte, el administrador crea un FDM para el SVV. Los

operadores de SVV simplemente tendrán que cargar dicho FDM.

### **Asociación de unidades**

Otra función fundamental del Servidor Entrenador Táctico es asociar entidades del mundo real a entidades del mundo virtual y garantizar que la información asociada sea correcta.

En la solución desarrollada, y para cada unidad, el emparejamiento se define como del mundo real al mundo virtual o del mundo virtual al mundo real. En el primer caso, la unidad sólo será gestionada desde el mundo real (por ejemplo su posición será capturada con un receptor GPS e insertada en el sistema de mando y control) y el SET se encargará de traducir las sucesivas posiciones de la unidad en movimientos coherentes en el mundo virtual del avatar correspondiente. En el segundo caso, la unidad será completamente simulada y el SET se encargará de enviar las actualizaciones de información del mundo virtual al nodo correspondiente en el mundo real, para que se incorporen en la dinámica de réplica habitual del sistema de mando y control.

Otros elementos que se han mapeado son las amenazas (representan elementos de naturaleza peligrosa) y los objetos (elementos genéricos) existentes en SIMACOP. En el sistema de información de mando y control C4ISR se pueden especificar y localizar en una ubicación, con su consiguiente representación en GIS. Cuando el servidor detecta que se inserta una entidad de estas características en el mundo real, instancia la entidad equivalente en el mundo virtual.

### ***Simulación de unidades***

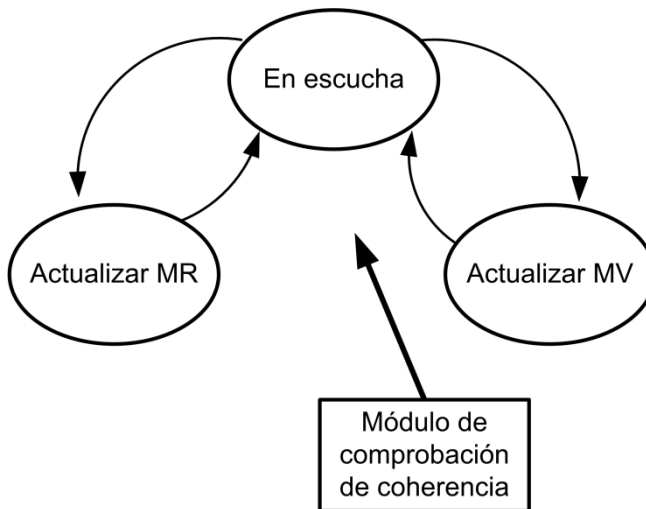
Un aspecto relevante del sistema desarrollado es permitir la simulación de unidades virtuales no gestionadas por ningún operador humano en el entrenamiento de operativos reales con sistemas de mando y control reales, tal como se identificó al analizar el estado del arte (concretamente, en el numeral 2.3.4).

Se hace evidente su utilidad, por ejemplo, en el entrenamiento de policías ante el gentío (miles de manifestantes simulados) o el entrenamiento de bomberos ante multitudes de heridos y víctimas simuladas en el simulacro de una catástrofe. En esta línea cabe destacar que se está trabajando en el área de movimientos coherentes y verosímiles de grandes números de avatares en un escenario, tanto para la gestión de unidades simuladas como para la gestión del movimiento coordinado de un conjunto de unidades mapeadas del mundo real al virtual (por ejemplo un pelotón de soldados moviéndose en formación). Se ha realizado un gran número de trabajos [BoDT99, ToBe04], principalmente para la industria de la animación infográfica [Gabb05, Mass15, Reyn87], pero ésta sigue siendo una línea de investigación abierta en su aplicación a la simulación y previsión del movimiento de grandes masas de avatares en áreas de gestión de emergencias y seguridad.

### ***Gestión de coherencia***

Hemos comentado que el principal elemento del SET es un proceso que al recibir alguna novedad mapea la información de un mundo a otro. Como se ha visto, hay muchas ocasiones en las que el mapeo

no es trivial y precisa el uso de elaborados algoritmos como en el caso de la gestión de unidades simuladas o en el de la generación de rutas óptimas. En cualquier caso, el funcionamiento del SET se puede describir con la máquina de estados mostrada en la Figura 43:



**Figura 43. Máquina de estados SET.**

Hay que destacar que para los flujos de información de MV a MR, a cada nodo del MR le insertamos la información que sería generada por él en condiciones normales (sin MV), de forma que garantizamos la consistencia de la réplica del MR. En el caso de los flujos de información del MR al MV esto no es necesario, pues el mundo virtual está un único repositorio de información y no en una base de datos distribuida.

El módulo de comprobación de coherencia básicamente trata de garantizar la consistencia entre las bases de datos del MR (distribuidas) y la base de datos o repositorio de información del MV

(en principio, y desde un punto de vista exterior al servidor de mundos virtuales, centralizada). Principalmente se debe supervisar:

- Consistencia de temporizadores.
- Creación de unidades.
- Movimiento de unidades de MR en MV.
- Movimiento de unidades simuladas.

La gestión de la coherencia es una línea de investigación importante en el área de desarrollo de motores de mundos virtuales. El entrenador táctico implementa actualmente técnicas básicas de gestión de la coherencia y la consistencia pero la incorporación de técnicas avanzadas se prevé como línea de trabajo futura.

#### **4.3.2. Cliente Entrenador Táctico**

El Cliente Entrenador Táctico (CET) es el subsistema cliente a través de cual interactúan los nodos de entrenamiento; incluye el sistema de mando y control C4ISR del mundo real y un cliente de mundo virtual.

En la interfaz de usuario (Figura 44), y posterior al procesamiento de los FDM cargados por el usuario, se visualizan y manejan unidades del mundo virtual, e igualmente se visualizan los avatares de las unidades que existen (y se mueven, etc.) en el mundo real.

La interfaz tiene dos componentes: una pantalla del C4ISR (izquierda) con toda la información pertinente: mensajería, unidades geo localizadas, objetos, etc.; y la representación equivalente de toda



esta información en el mundo virtual (derecha). Ambos sistemas son completamente operativos como si estuvieran siendo utilizados fuera del CET: si pulsamos sobre SIMACOP podremos acceder a todas sus funcionalidades; lo mismo con el cliente OLIVE.

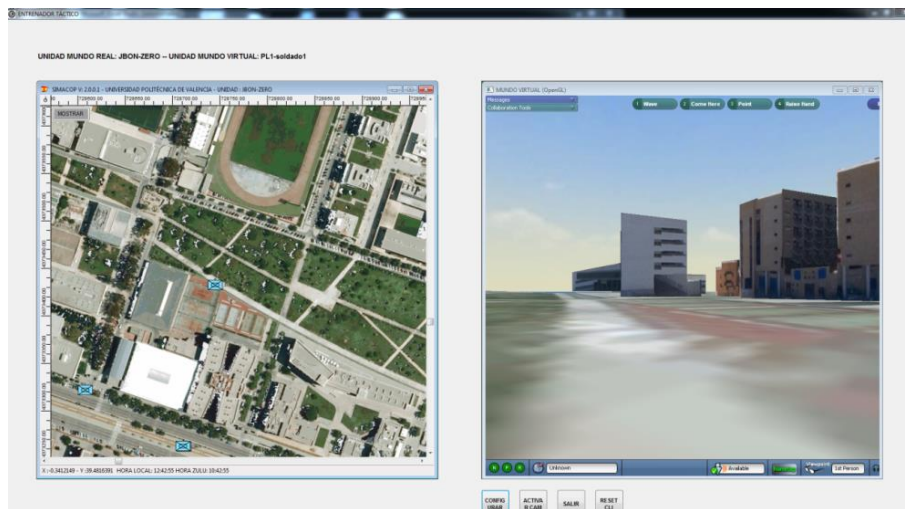


Figura 44. CET. Pantalla principal.

Hay una conexión directa entre la posición geográfica, basada en GPS, y la representación en el mundo virtual, de manera que cuando cualquier entidad ejecuta alguna acción, se transmite la información usando el formato intermedio MPEG-V. Adicionalmente, se puede configurar la estructura de mando y el escenario de entrenamiento.

El sistema está diseñado para adaptarse a las características de pantalla del equipo en el que se ejecuta, adaptando la interfaz y maximizando el área visible de ambos mundos.

La Figura 45 muestra un caso de uso del CET. En SIMACOP se inserta un objeto y éste es transferido al mundo virtual, donde en

todo momento se observa el mundo desde el punto de vista del avatar correspondiente a una de las unidades del mundo real.

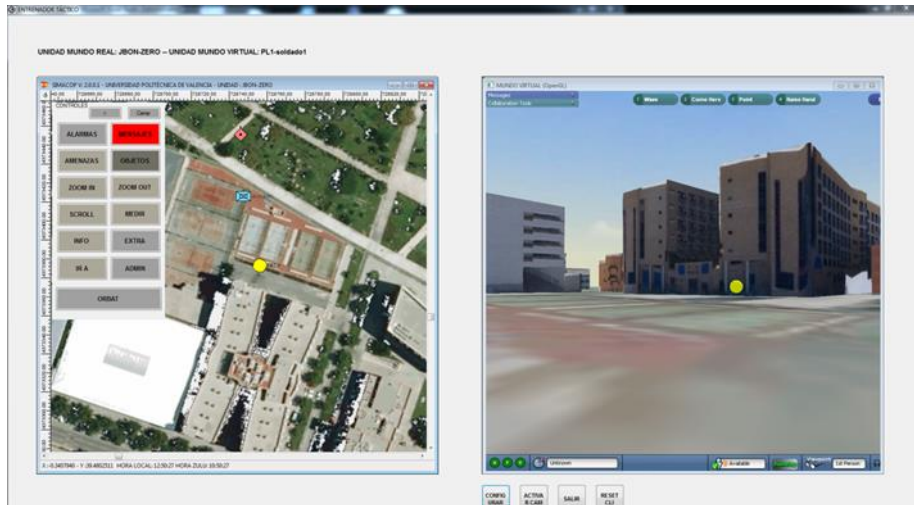


Figura 45. Transferencia de objetos del MR al MV.

Cabe destacar que el sistema de mando y control, SIMACOP, dispone de un reproductor de vídeo. Convenientemente configurado el FDM, el sistema puede solicitar el vídeo propio de cada unidad al SVV (sección 4.3.3) y éste se lo entregará mediante *multicast* (si la red subyacente lo permite). En la Figura 46 podemos ver en el CET el cliente de OLIVE (parte derecha de la interfaz) mirando hacia otras unidades y mientras tanto en SIMACOP (a la izquierda), se va reproduciendo, en tiempo real, el vídeo de las unidades observadas por el cliente.

#### 4.3.3. Servidor de Vídeo Virtual

El Servidor de Vídeo Virtual (SVV), cuya arquitectura se mostró anteriormente en la Figura 20, es el sistema de gestión, distribución y

reproducción de los flujos de vídeo generados en el sistema, de fuentes tanto reales como virtuales. Es una aplicación independiente respecto a la pasarela de interconexión, y su misión es la de gestionar los distintos flujos de vídeo que generan los sensores de los sistemas CET y sensores independientes reales, y redistribuirlos a los clientes que lo soliciten.

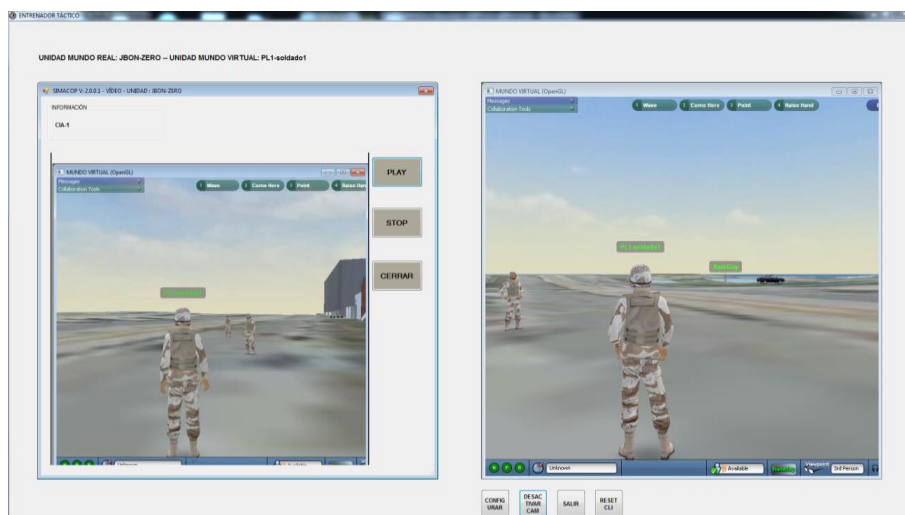


Figura 46. Vídeo del MV en el MR

Cuando un sensor de vídeo de un CET arranca, comprueba cuál es su SVV asociado y empieza a enviar vídeo directamente a la IP del SVV. Esta IP la encuentra en su FDM. El SVV tiene la capacidad de determinar qué flujos en vivo van a estar disponibles para el resto de nodos y qué flujos van a ser grabados (los flujos se graban en formato MPEG-4 de forma que pueden ser vistos posteriormente utilizando cualquier reproductor de vídeo). Una vez se han habilitado los flujos de vídeo, podrán ser visualizados en el SVV. La Figura 47 muestra un ejemplo de visualización, con 4 flujos de vídeo.

Es posible seleccionar la reproducción de cámaras en vivo o visualizar grabaciones ya existentes en el repositorio de vídeo. También es posible configurar el número de flujos a visualizar y su disposición y seleccionar el tamaño de los *displays* de visualización. El sistema se adapta a las características de la pantalla del equipo en el que esté ejecutándose, ajustando su tamaño a la resolución.

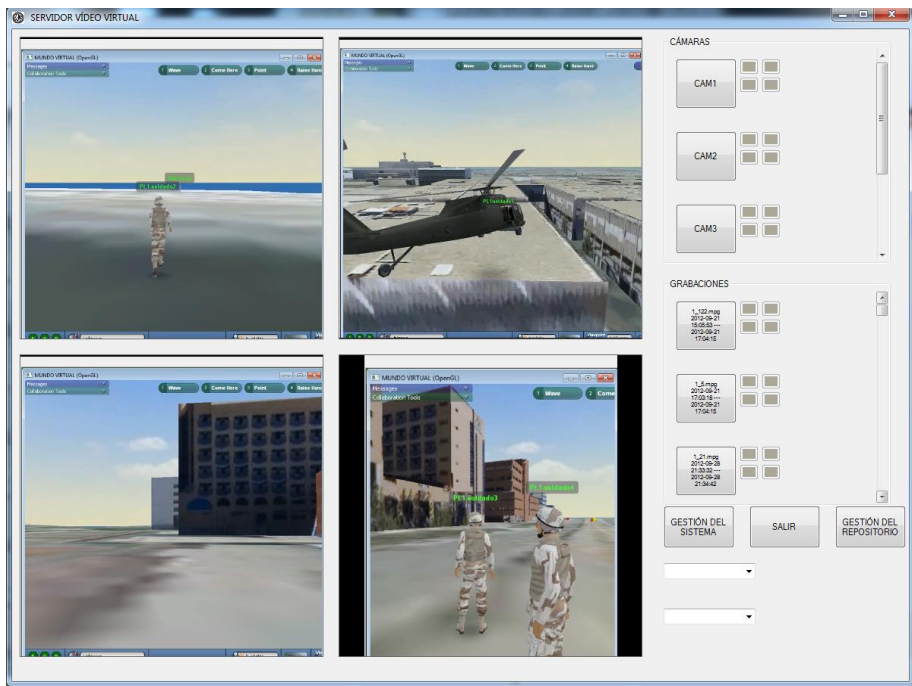


Figura 47. SVV. Pantalla principal con 4 displays.

Por otra parte, el servidor de vídeo desacopla productores de consumidores y una consecuencia directa de este desacoplo es la reducción del ancho de banda consumido. El sistema recibe un flujo de vídeo por cada fuente y lo retransmite usando *multicast*, disminuyendo el número total de flujos respecto al caso en el que únicamente hubiera productores y consumidores.

## 4.4. Comunicación entre nodos

La comunicación se basa en servicios web vía TCP/IP, como se comentó anteriormente en la sección 3.5.

Inicialmente se había planteado usar C# y .NET para la implementación del paradigma de comunicación, y las primeras pruebas se hicieron en este entorno [APEP15]. Sin embargo, a medida que se avanzaba en la tesis y por algunos obstáculos encontrados durante el proceso de desarrollo, fue necesario explorar otras alternativas.

La implementación final del servicio web utiliza JSON para el encapsulamiento de datos, y ha sido desarrollada usando *Ruby on Rails*. Hay dos razones principales que justifican esta decisión:

- La transmisión de datos usando JSON es mucho más eficiente [Wang11] y reduce la carga considerablemente. Si además se implementase compresión en el servidor web, el envío de datos sería aún más eficiente.
- Se reducen costos, ya que se utiliza software de código abierto.
- El desarrollo y pruebas son más sencillos, lo que a su vez facilitaría futuras ampliaciones del sistema (por ejemplo, si se quisiese introducir un nuevo formato de representación para validar los datos, además de MPEG-V).

Una vez hecha esta aclaración, veamos el funcionamiento del servicio:

```

=> Booting WEBrick
=> Rails 4.2.4 application starting in development on http://0.0.0.0:3000
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server
[2015-10-25 01:00:56] INFO WEBrick 1.3.1
[2015-10-25 01:00:56] INFO ruby 2.2.2 (2015-04-13) [x86_64-linux]
[2015-10-25 01:00:56] INFO WEBrick::HTTPServer#start: pid=1375 port=3000

Started POST "/api/v1/convert/mpegv" for 10.0.2.2 at 2015-10-25 11:29:42 +0100
Cannot render console from 10.0.2.2! Allowed networks: 127.0.0.1, ::1, 127.0.0.0/127.255.255.255
Processing by Api::V1::ConversionController#to_mpegv as JSON
  Parameters: {"SystemElements":{"OliveAssoc":{"OliveAssocId":"42","olvid":"1016825","oliveName":"CIA-1","smcId":"2","smcName":"CIA-1"},"PoolSimBom":{"Avoid":"1011437","PoolSimBomId":"2"},"PoolSimMil":{"Avoid":"1011407","PoolSimMilId":"2"},"PoolSimPol":{"Avoid":"1011452","PoolSimPolId":"2"},"PoolSimSan":{"Avoid":"1011422","PoolSimSanId":"2"},"PoolSimulated":{"PoolSimulatedId":"2","RangeFree":"0","RangeStart":"1011417","SimulatedRangeType":"1"},"Position":{"DateTime":"2012-09-10 09:31:35","PosId":"1","PosXYZ":{"x":"0.00000000","y":"0.00000000"},"SimulatedUnits":{"InitialPosXYZ":{"InitialX":"0","InitialY":"0","InitialZ":"0"},"SimulatedUnitID":"1","oliveName":"military"},"SimulatedUnitsType":{"SimulatedUnitName":"sanitario","SimulationUnitTypeId":"2"},"SimulationBehaviourType":{"BehaviourTypeName":"flockU1","SimulationBehaviourTypeId":"2"},"Threat":{"DateTime":"2012-10-08 12:47:52","Origin":"1","PosXYZ":{"x":"55699","y":"55166"},"ThreatId":"21","ThreatType":"1"}}}=>nil}
Completed 200 OK in 52ms (Views: 0.1ms | ActiveRecord: 0.0ms)

```

Figura 48. Envío de datos con JSON.

El servicio web constantemente monitorea el bus de datos del entrenador táctico. Cuando se detecta que alguna entidad del sistema inserta datos, éstos son capturados y enviados al servidor web, mediante JSON, en un documento XML. A continuación son convertidos a MPEG-V de acuerdo al modelo de datos propuesto en la sección 4.2, utilizando la librería de Ruby *Nokogiri* [PDNA15] para generar el XML que será entregado al nodo destino de sistema. En la Figura 48 se muestra un ejemplo del envío de datos procesados.

El servicio web se ha implementado de manera modular y como una capa adicional a los servicios creados durante la implementación del entrenador táctico, de forma que es totalmente independiente y reutilizable. Para la implementación y pruebas se configuró un

servidor web básico, así como varios clientes. Sus características se muestran en la Tabla 1.

**Tabla 1. Configuración del servidor web y clientes.**

	CPU	RAM	O.S.	HDD
Servidor	Intel Core i3-4005U @ 1.7 GHz	512 Mb	Ubuntu 14.4 LTS	4 Gb SATA
Clientes	Intel Core i3-4005U @ 1.7 GHz	1 Gb	Windows XP Prof. SP2 x32	40 Gb SATA

La Figura 49 muestra el servicio en funcionamiento. En este caso, una entidad del MR ha cambiado su posición y se transmite toda su información asociada para actualizarla en el avatar que le corresponde en el MV.

Vale la pena recordar que los documentos XML generados con el servicio web implementado no sólo son válidos conforme a la normativa MPEG-V, sino que además cumplen las recomendaciones de la W3C.

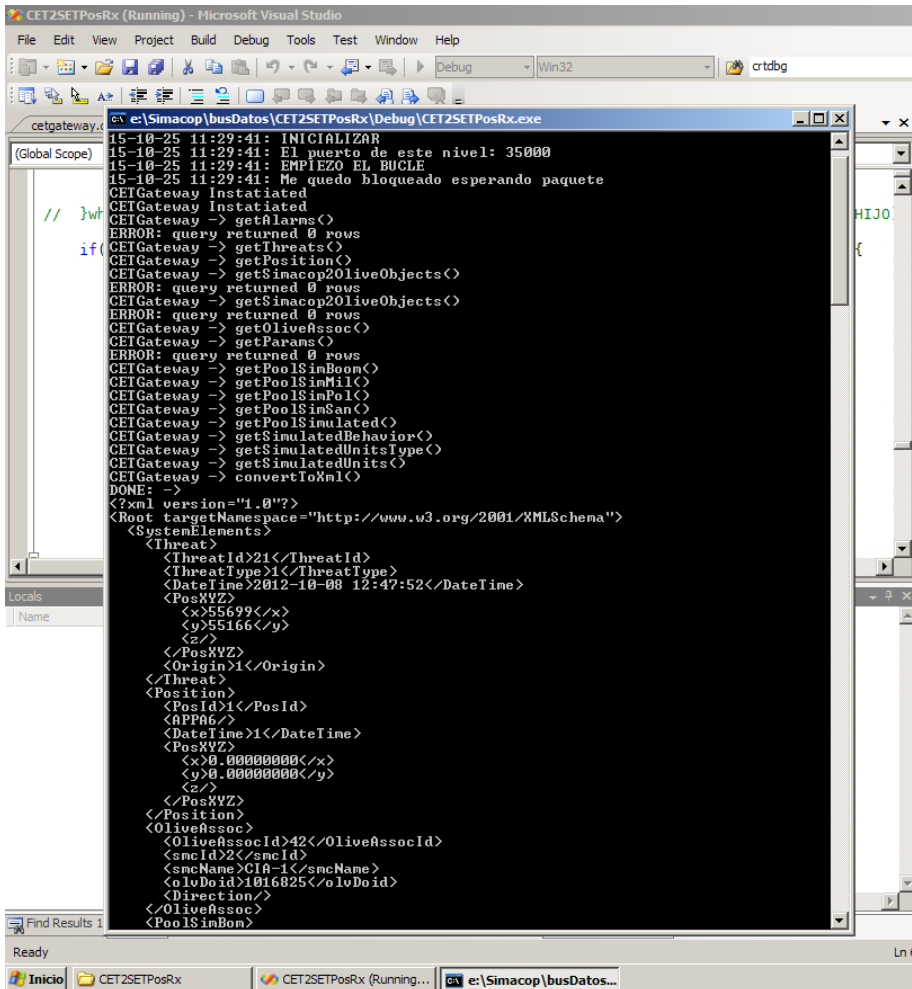


Figura 49. Ejemplo funcionamiento servicio web.

## 4.5. Bibliografía

[APEP15] ARDILA, LAURA; PÉREZ-LLOPIS, ISRAEL; ESTEVE, MANUEL; y PALAU, CARLOS E.: Interoperable architecture for joint real/virtual training in emergency management using the MPEG-V standard. En: *Computer Standards & Interfaces* vol. 41 (2015), pp. 39-55.



[BoDT99] BONABEAU, ERIC; DORIGO, MARCO; y THERAULAZ, GUY: *Swarm intelligence: from natural to artificial systems*: Oxford University Press, Inc., 1999 — ISBN 0-19-513159-2.

[Gabb05] GABBAI, JONATHAN M. E.: *Complexity and the Aerospace Industry: Understanding Emergence by Relating Structure to Performance using MultiAgent Systems*, Tesis doctoral, University of Manchester, 2005.

[Mass15] MASSIVE: <http://www.massivesoftware.com/>.

[PDNA15] PATTERSON, AARON; DALESSIO, MIKE; NUTTER, CHARLES; ARBEO, SERGIO; ET AL.: <http://www.nokogiri.org/>.

[Reyn87] REYNOLDS, CRAIG W.: Flocks, herds and schools: A distributed behavioral model. En: *ACM SIGGRAPH Computer Graphics*. vol. 21 : ACM, 1987 — ISBN 0-89791-227-6, pp. 25-34.

[ToBe04] TOPAZ, CHAD M. y BERTOZZI, ANDREA L.: Swarming Patterns in a Two-Dimensional Kinematic Model for Biological Groups. En: *SIAM Journal on Applied Mathematics* vol. 65, Society for Industrial and Applied Mathematics (2004), Nr. 1, pp. 152-174.

[Wang11] WANG, GUANHUA: Improving Data Transmission in Web Applications via the Translation between XML and JSON. En: *2011 Third International Conference on Communications and Mobile*

*Computing*, 2011 — ISBN 978-1-61284-312-4, pp. 182-185.

[Wor15] WORLD WIDE WEB CONSORTIUM (W3C):  
<http://www.w3.org/standards/xml/schema>.

# Capítulo 5

## Validación

### 5.1. Introducción

En este capítulo se describirá la plataforma implementada como banco de pruebas y los resultados obtenidos. Se demuestra la viabilidad del sistema y se evalúa que el rendimiento es adecuado, validando así la arquitectura propuesta en esta tesis doctoral.

### 5.2. Validación del sistema

En aplicaciones de mando y control, y en particular en las relacionadas con gestión de emergencias, el adecuado uso de recursos es un aspecto crítico y de vital importancia. Por esto, se analizó el consumo de CPU y ancho de banda de cada componente del sistema (SET, CET y SVV), para diferentes configuraciones y una ventana de tiempo de 300 segundos.

La Figura 50 muestra la estructura de red de la plataforma implementada como banco de pruebas. El sistema consta de un Servidor Entrenador Táctico, cuatro Clientes Entrenador Táctico, un

Servidor de Vídeo Virtual, y un sistema de mando y control del mundo real.

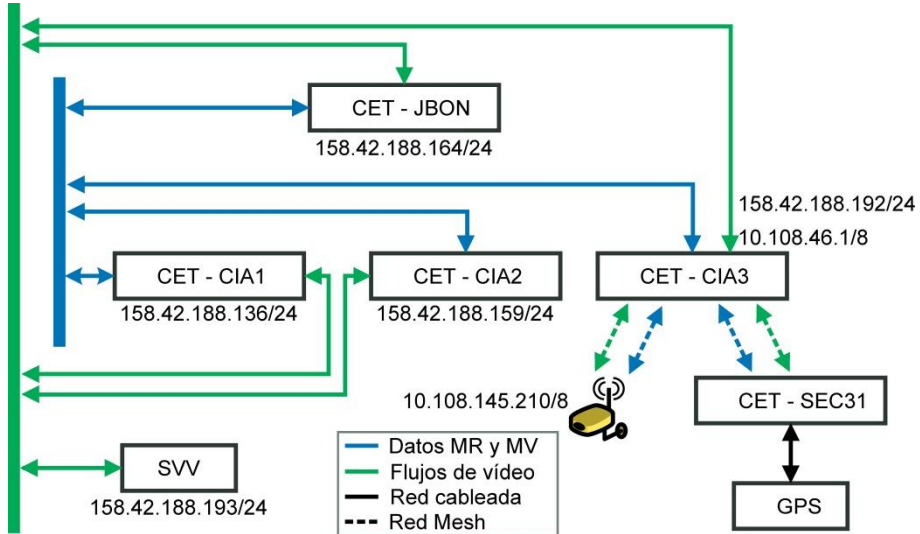


Figura 50. Estructura de red de la plataforma de pruebas.

Las características de los equipos utilizados se detallan en la Tabla 2. Cabe mencionar que no se han considerado significativas para el análisis las variaciones que puedan surgir por las diferentes características de cada equipo y por tanto han sido obviadas; si bien podría ser interesante, en etapas futuras, incluir parámetros como la velocidad de transmisión de datos para diferentes interfaces, o la influencia que otros procesos propios del sistema operativo podrían tener en las mediciones.

### 5.2.1. Consumo de CPU Servidor Entrenador Táctico

Se estudió el consumo de CPU en función del número de clientes, variando de uno (Figura 51a) a cinco (Figura 51b). Puede verse,

como es de esperar, que a mayor número de clientes, mayor será el consumo de CPU.

Aplicando una regresión lineal con mínimos cuadrados a los resultados experimentales, se pudo comprobar que el consumo de CPU aumenta de manera proporcional al número de clientes, y que para 25 clientes sería de un 49,89%. La Figura 51c muestra los resultados experimentales y el uso estimado.

**Tabla 2. Equipo utilizado en la plataforma de pruebas.**

	CPU	RAM	O.S.	HDD
CET – JBON	Intel Core i3-2100 @ 3.1 GHz	12 Gb	Windows 7 64 bits	1 Tb SATA
CET – CIA1	Intel Core i7-2675QM @2.2 GHz	8 Gb	Windows 7 64 bits	1 Tb SATA
CET – CIA2	Intel Core i3-2100 @ 3.1 GHz	4 Gb	Windows 7 64 bits	1 Tb SATA
CET – CIA3	Intel Core 2 duo L9300 @1.6 GHz	2 Gb	Windows XP Prof. SP2 x32	160 Gb
CET-SEC-31	CPU: Intel Atom Z530 @1.6 GHz	2 Gb	Windows 7 64 bits	60 Gb
SVV	Intel Core i3-2100 @ 3.1 GHz	12 Gb	Windows 7 64 bits	1 Tb SATA

### **5.2.2. Consumo de ancho de banda Servidor Entrenador Táctico**

Se analizó el consumo de ancho de banda en función del número de clientes en la pasarela para dos casos:

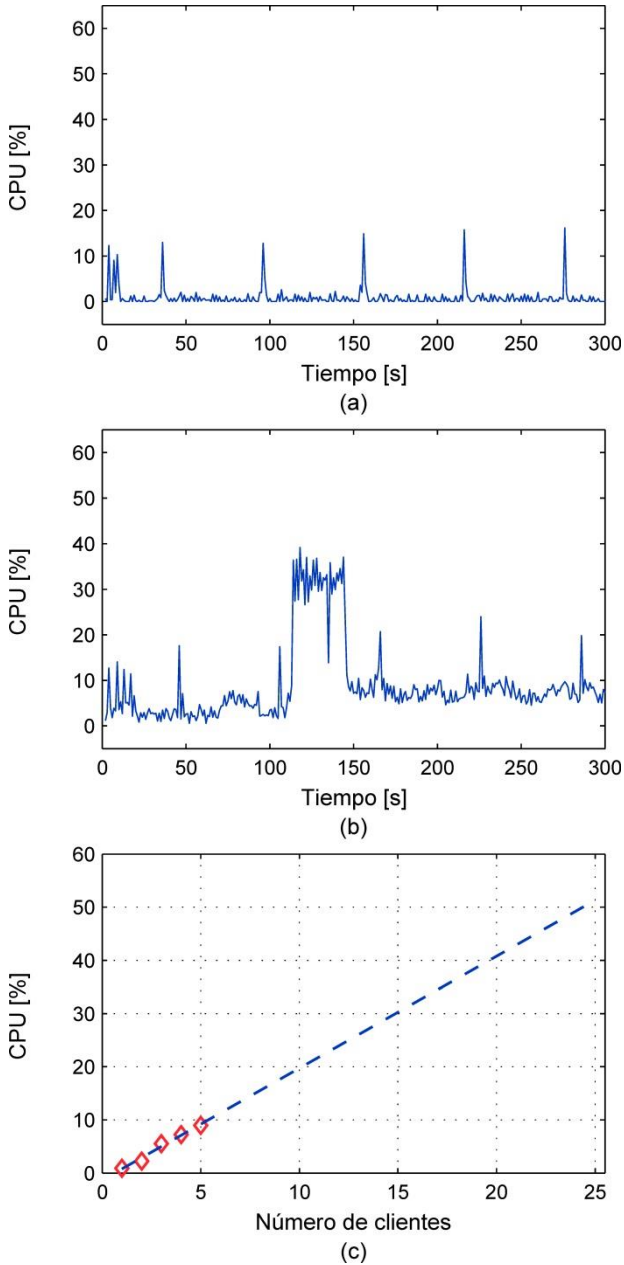


Figura 51. SET. Consumo de CPU. (a) 1 cliente. (b) 5 clientes. (c) Evolución estimada del consumo de CPU en función del número de clientes.

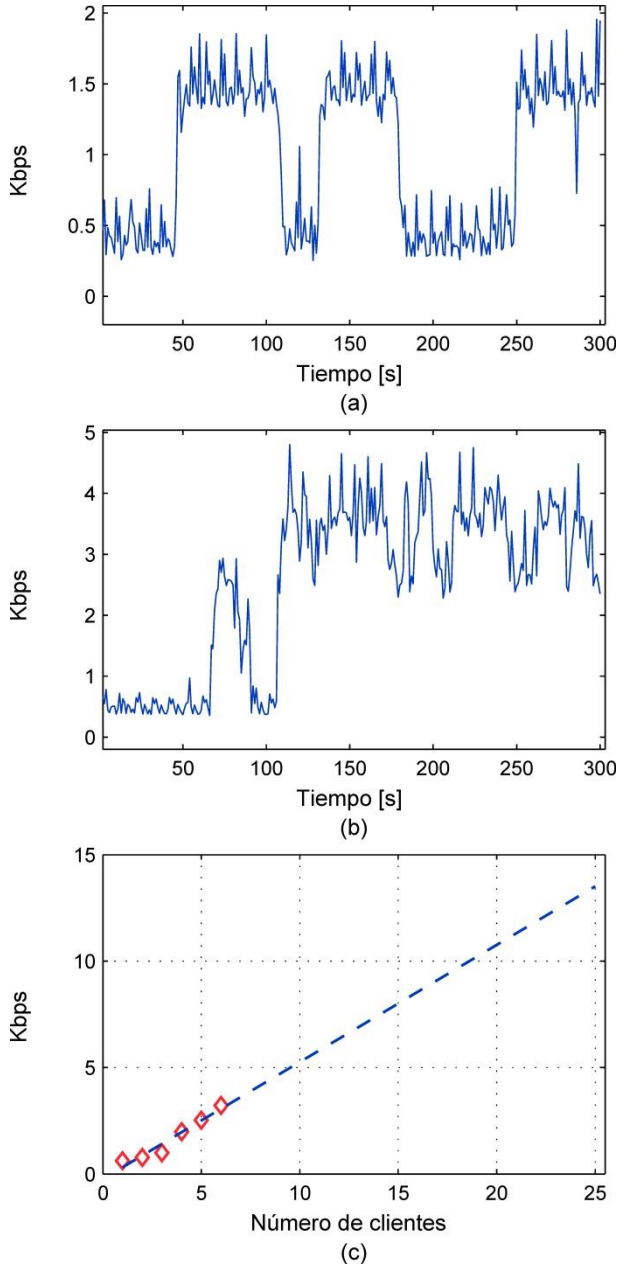
- Intercambio de datos entre el mundo real y mundos virtuales.
- Tráfico de información para el mantenimiento del estado entre nodos del mundo virtual.

En primer lugar, se estudió el uso de ancho de banda cuando se intercambia información entre la pasarela y los nodos CET variando de uno a cinco clientes. Todo el tráfico es TCP.

Se observó que el consumo sigue un ciclo con dos estados, uno de mayor actividad de réplica y otro con menos. En el caso de tres clientes el consumo promedio es de 0.5 Kbps y los estados se pueden diferenciar claramente (Figura 52a), mientras que para cinco clientes las variaciones de estado son menos evidentes y se consumen en promedio 3.32 Kbps (Figura 52b).

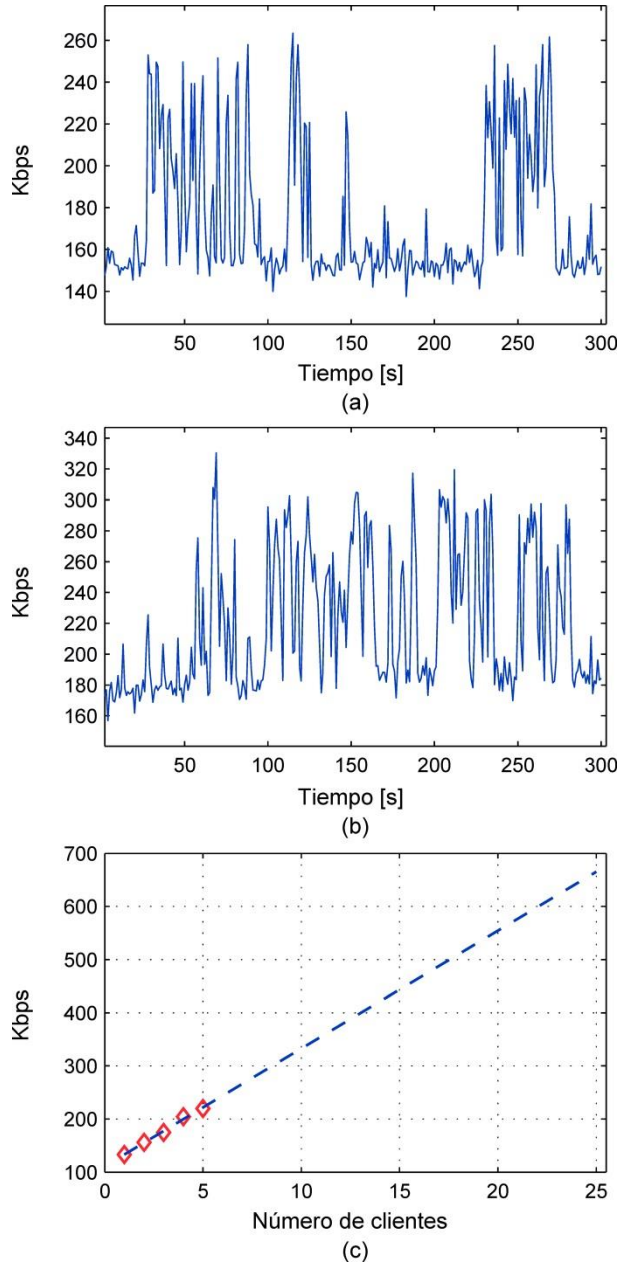
Basados en los resultados, y para predecir el comportamiento del sistema para un número mayor de nodos, se fijó un rango operativo de 25 clientes para el cual el sistema no colapsará y dentro del que puede asumirse una relación lineal. Entonces, extrapolando los valores experimentales, y aplicando nuevamente el método de regresión lineal con mínimos cuadrados, se ha obtenido el comportamiento estimado. Se muestra en la Figura 52c.

A continuación, se analizó el tráfico exclusivamente del lado virtual del sistema para uno a cinco clientes. En la Figura 53 puede verse que, en contraste con el consumo de ancho de banda en la pasarela donde se distinguían dos estados, en este caso el tráfico no sigue un ciclo determinado y se presentan algunos picos de consumo.



**Figura 52. SET. Uso ancho de banda pasarela MR-MV. (a) 3 clientes asociados. (b) 5 clientes. (c) Estimación del consumo en función del número de clientes.**





**Figura 53. SET. Ancho de banda consumido en intercambio entre mundos virtuales. (a) 3 clientes. (b) 5 clientes. (c) Uso estimado en función del número de clientes.**

A partir de los resultados, se pudo concluir que el consumo de ancho de banda aumenta con un factor aproximado de 1.2 a medida que aumenta el número de clientes. La Figura 53c muestra la evolución estimada del consumo para 25 clientes.

Finalmente, se ha hecho una comparativa de los dos casos. Se comprueba que el consumo de ancho de banda en la pasarela aumentará en menor grado que cuando se intercambian datos entre mundos virtuales. En la Figura 54 se comparan los resultados; se ha usado escala logarítmica para apreciar mejor las diferencias.

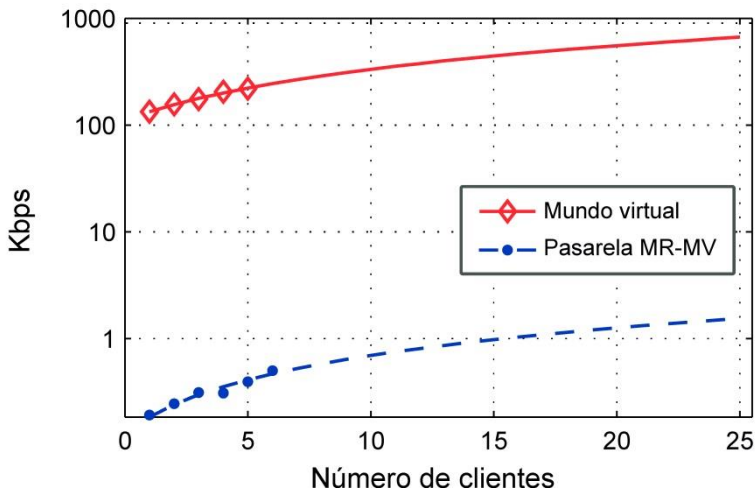


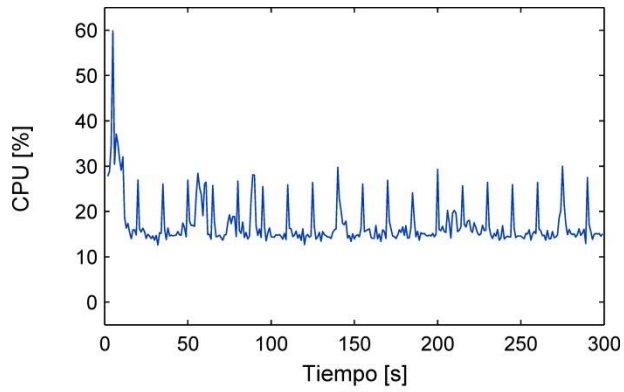
Figura 54. SET. Comparación consumo estimado ancho de banda pasarela MR-MV vs. MV en función del número de clientes.

### 5.2.3. Consumo de CPU Cliente Entrenador Táctico

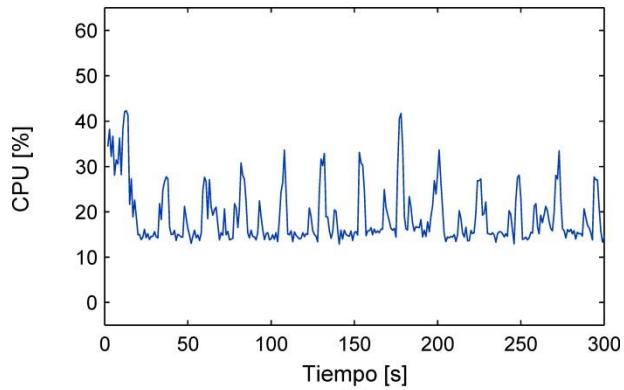
Se analizó el consumo de CPU para un único cliente y tres casos diferentes:

- Cuando no está conectado a la pasarela ni envía vídeo (Figura

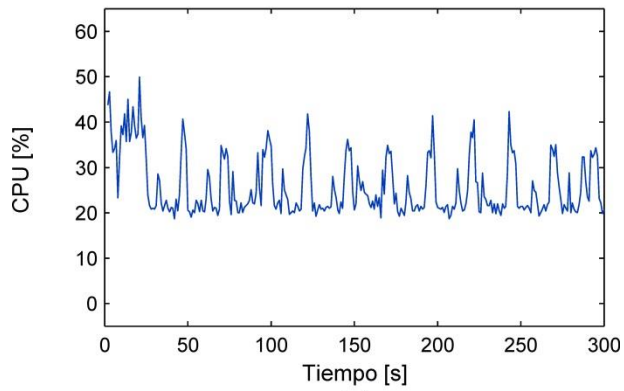
55a). Se observó un consumo de CPU promedio de 17.12%.



(a)



(b)



(c)

Figura 55. CET. Consumo de CPU para diferentes estados.

- Cuando está conectado a la pasarela sin enviar vídeo (Figura 55b). En este caso, el consumo aumenta ligeramente con respecto al caso anterior, con un valor medio de 19.2%.
- Cuando está conectado la pasarela y envía vídeo (Figura 55c). Como es de esperar, el mayor consumo se presenta en este caso y alcanza un valor promedio de 25.8%.

#### **5.2.4. Consumo de ancho de banda Cliente Entrenador Táctico**

Se definieron dos escenarios:

- Cuando sólo hay réplica de información en el mundo real. Se hicieron mediciones para 3 nodos (Figura 56a) y 5 nodos (Figura 56b) conectados al sistema. Podemos observar cómo la réplica de información del mundo real introduce picos de consumo variando entre 20 y 180 Kbps.

En la Figura 56c se predice cómo evolucionará el consumo de ancho de banda a medida que el número de clientes del sistema aumenta. Podemos ver que el comportamiento será muy similar al observado para el SET dentro del mismo rango operativo.

- Cuando hay intercambio de información para la gestión del mundo virtual. Para este escenario se evaluaron los mismos casos que para el consumo de CPU del numeral anterior, y para un cliente cuando: (i) no está conectado a la pasarela ni enviando vídeo; (ii) hay conexión con la pasarela pero no transmisión de vídeo; y (iii) está conectado a la pasarela y enviando vídeo.

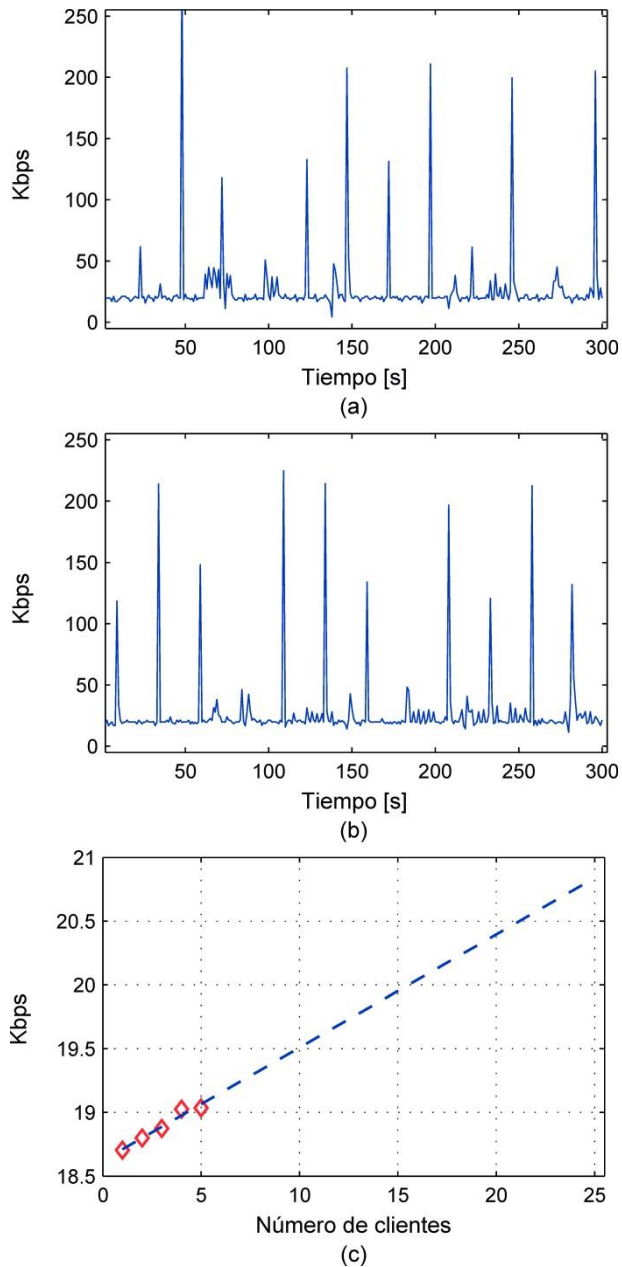


Figura 56. CET. Consumo ancho de banda. Réplica MR. (a) 3 clientes. (b) 5 clientes. (c) Evolución estimada en función del número de clientes.

Los resultados se muestran en la Figura 57. Se observan más variaciones en el consumo que para el primer escenario, si bien se mantienen dentro del mismo rango de valores. Al igual que se observó en el SET, cuando hay transmisión de vídeo el uso de ancho de banda aumentará.

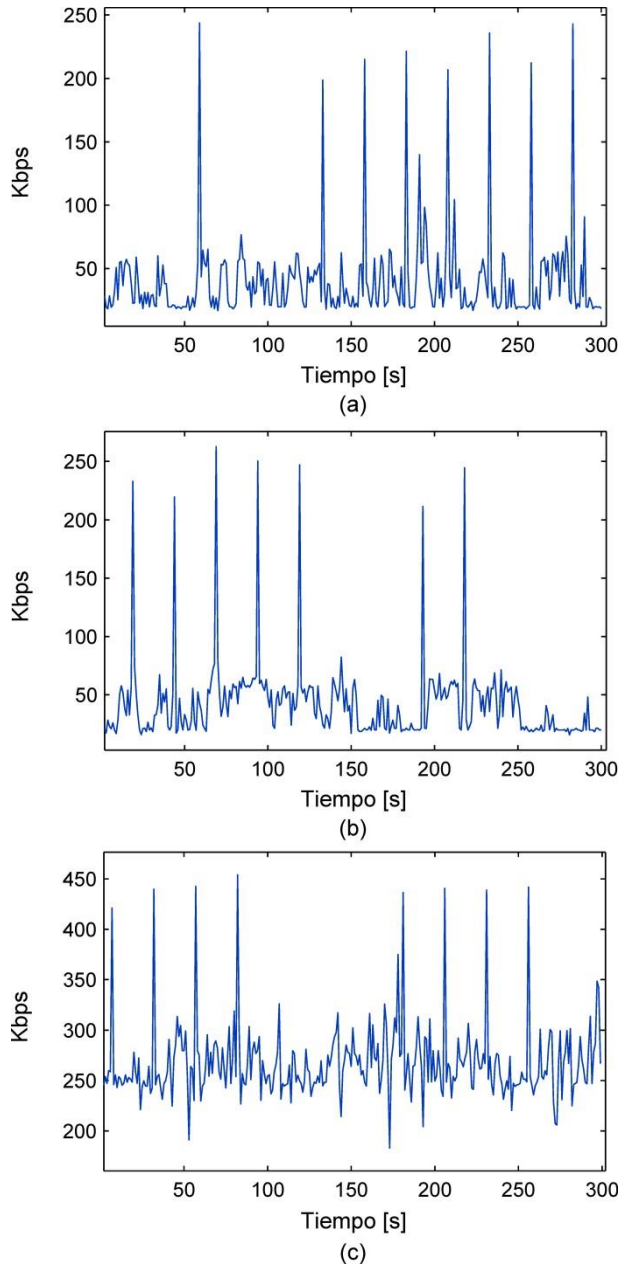
### 5.2.5. Consumo de CPU Servidor de Vídeo Virtual

En primer lugar se analizó el consumo de CPU para uno a cuatro flujos de vídeo entrantes y dos escenarios:

- Transmisión sin grabación.
- Transmisión y grabación.

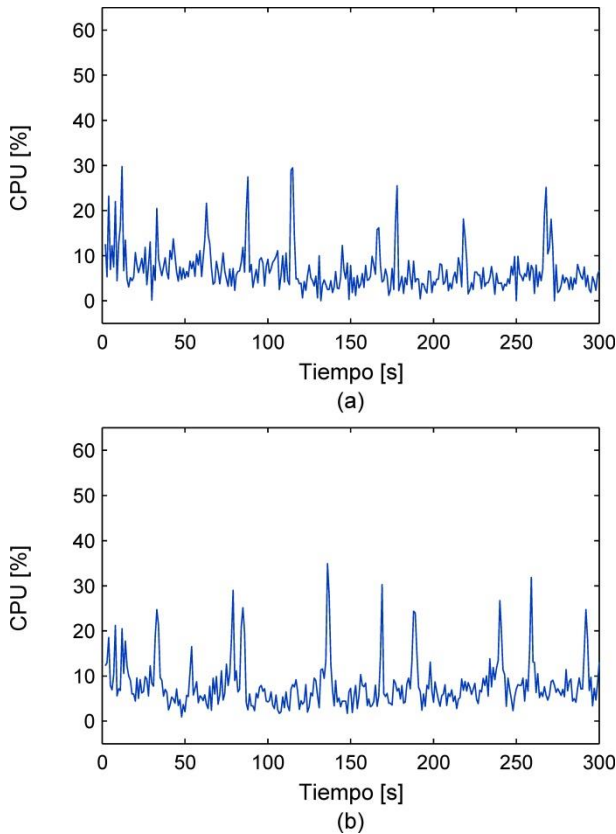
**Tabla 3. Consumo de CPU en el SVV para uno a cuatro flujos de vídeo.**

Escenario	Flujos de vídeo	Valor medio [%]	Valor máximo [%]	Varianza
Transmisión sin grabación	1	5.4979	30.6	25.21
	2	6.264	27.432	18.601
	3	5.296	28.239	22.620
	4	6.786	29.760	24.357
Transmisión y grabación	1	5.7424	36.42	33.35
	2	6.998	29.019	31.447
	3	7.098	27.820	28.683
	4	8.028	34.869	28.987



**Figura 57. CET. Consumo de ancho de banda para la gestión de MV. (a) Sin conexión ni vídeo. (b) Conexión a pasarela sin vídeo. (c) Conexión a pasarela y transmisión de vídeo.**

Una observación interesante es que el consumo promedio de CPU cuando no hay grabación siempre es menor, independientemente del número de clientes; pero no siempre ocurre lo mismo con el valor máximo. Esto se debe a que como las muestras se han tomado con ejecuciones reales del sistema pueden existir valores puntuales superiores debido a las características de la entrada. Por su parte, la varianza es mayor cuando hay grabación para todos los casos, debido a la mayor dispersión que introduce cada proceso de grabación en la carga de CPU.



**Figura 58. SVV. Consumo de CPU para 4 flujos de vídeo. (a) Transmisión de vídeo. (b) Transmisión y grabación de vídeo.**



En la Figura 58 se muestran los resultados para cuatro flujos de vídeo y para los dos casos.

Finalmente, y siguiendo el mismo procedimiento que para los demás componentes del sistema entrenador táctico, se ha hecho una regresión lineal con mínimos cuadrados para estimar la evolución en el consumo de CPU en función del número de flujos de vídeo para ambos casos. Se muestra en la Figura 59.

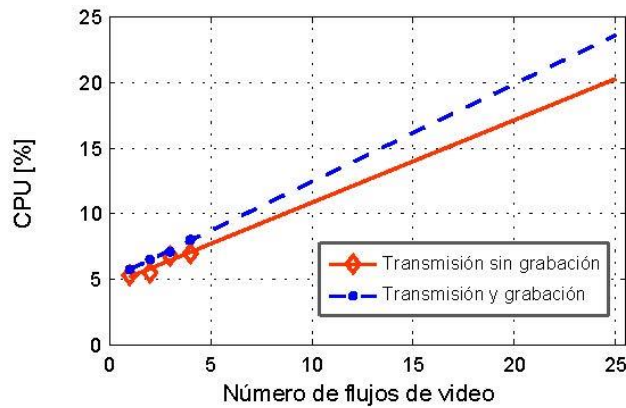


Figura 59. VVS. Estimación de evolución consumo de CPU para transmisión de vídeo con y sin grabación en función del número de flujos.

### 5.2.6. Consumo de ancho de banda Servidor de Vídeo Virtual

Por último, y tal como se hizo para SET y CET, se midió el ancho de banda consumido por el SVV. Nuevamente se varió el número de flujos entre uno (Figura 60a) y cuatro (Figura 60b), observando que para un único flujo hay bastantes variaciones en el consumo, mientras que a medida que aumenta el número de flujos éste tiende a ser más estable. Asimismo, se hizo la predicción de consumo en función del número de flujos de vídeo (Figura 60c).

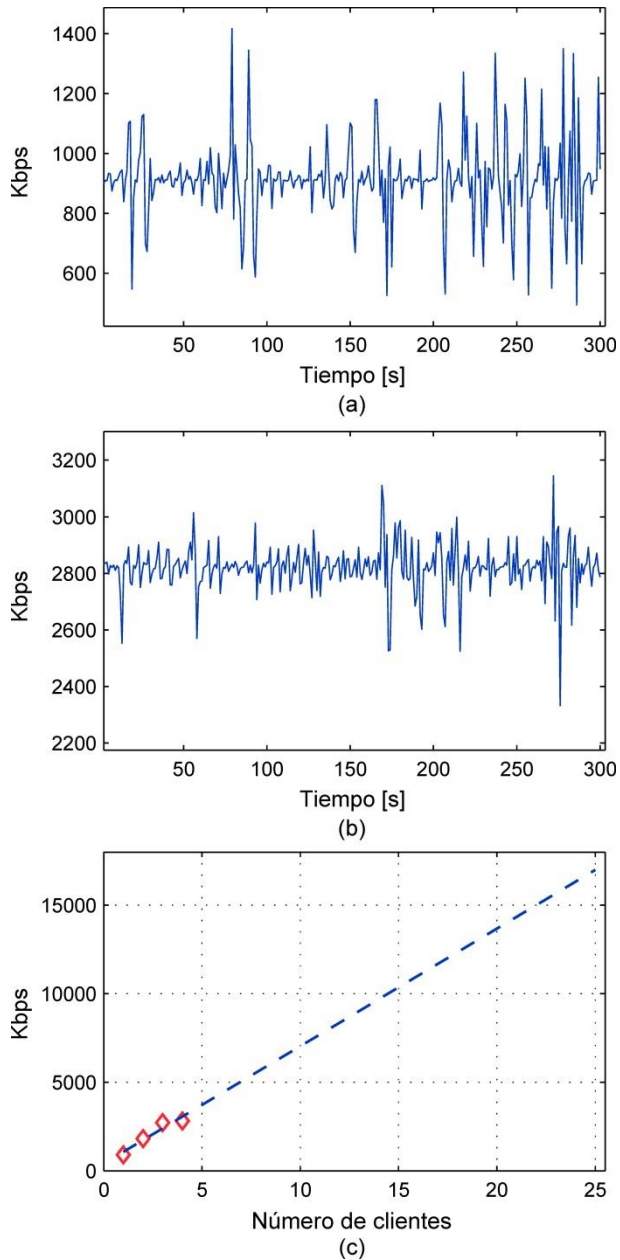


Figura 60. SVV. Consumo de ancho de banda. (a) 1 flujo de vídeo. (b) 4 flujos de vídeo. (c) Estimación consumo en función del número de flujos.

# Capítulo 6

## Conclusiones y desarrollos futuros

### 6.1. Conclusiones

Una vez desarrollada la tesis doctoral, se extraen las siguientes conclusiones:

- Se ha hecho una extensa revisión bibliográfica de los sistemas de mando y control, los mundos virtuales y el uso de ambos para entrenamiento de operativos de gestión de emergencias, así como un concienzudo estudio del estándar MPEG-V (ISO/IEC 23005). A partir de esto, se han identificado las brechas en el estado del arte y verificado la pertinencia de la presente tesis doctoral.
- Con esta tesis se hacen dos contribuciones muy importantes al estado del arte: en primer lugar, se propone una arquitectura para interconectar un C4ISR para gestión de emergencias con un mundo virtual que simula de forma precisa el mundo real en que opera el C4ISR. Segundo, se ha definido e implementado un modelo de datos, acorde al formato de representación definido por el estándar MPEG-V, que contribuye a garantizar la

interoperabilidad.

- Para el diseño de la arquitectura se han considerado las arquitecturas de mando y control que reflejan dos dimensiones: la estructura organizacional y la estructura de comunicaciones. Por otra parte, si bien el servidor de simulación es centralizado, la naturaleza distribuida del sistema permite simular en tiempo real la misma crisis en computadores separados geográficamente. Las interacciones de cada participante, ya sea en un mundo virtual o el real, son transparentes al sistema, con lo que se mantiene la coherencia durante toda la simulación.
- La arquitectura propuesta implementa módulos distribuidos e independientes pero compatibles y colaborativos entre sí, haciendo posible interconectar múltiples sistemas de mando y control con mundos virtuales y manejando adecuadamente el formato de datos nativo para adaptarlo a la normativa MPEG-V y garantizar interoperabilidad.
- El sistema es escalable, admitiendo cientos de usuarios y objetos, tanto reales como virtuales, que pueden interactuar entre sí, pero disminuyendo los costes asociados que implicaría llevar a cabo el mismo ejercicio de entrenamiento en un escenario real.
- Se ha implementado el mecanismo de comunicación utilizando JSON para la transmisión de datos y *Ruby on Rails* para el desarrollo de los procedimientos de traducción y adaptación de datos del formato nativo de cada aplicación al formato intermedio MPEG-V. Con esta aproximación se consigue una comunicación

altamente eficiente, disminuyendo la verbosidad y carga de transmisión de datos. Además, al no ser necesaria la adquisición de licencias, se reducen los costes de desarrollo y publicación asociados.

- El uso de entornos incluyendo mundos virtuales para entrenar en gestión de emergencias permite reducir de forma importante los costes de realización de los ejercicios de entrenamiento. Deja de ser imprescindible movilizar todas las unidades ya que pueden estar distribuidas en diversas ubicaciones geográficas y, además, se evitan posibles demoras ocasionadas por factores externos (p.e. gestión de permisos o trámites burocráticos).
- El uso de un sistema como el propuesto permite realizar un entrenamiento de operativos regular y continuo facilitando la planificación de gestión de crisis, así como la familiarización del personal con los procedimientos, roles y responsabilidades que les corresponden. Se contribuye a mejorar la eficacia de actuación y a reducir el tiempo de respuesta.
- Se han hecho mediciones del uso de recursos y se ha evaluado que la comunicación entre nodos sea eficiente, así como que los datos intercambiados entre ellos, a través de la pasarela de interconexión, sean consistentes y coherentes. Los resultados favorables demuestran que el sistema tiene un rendimiento satisfactorio y el uso de recursos es adecuado. Además, validan la arquitectura del sistema y confirman su viabilidad.
- El abanico de posibilidades al utilizar una arquitectura como la

propuesta para el entrenamiento conjunto real y virtual en sistemas de mando para control de emergencias es inmenso. Se consigue armonizar los procedimientos entre organizaciones y aumentar la flexibilidad en la configuración de ejercicios de entrenamiento, lo que permite que los operativos practiquen diversos protocolos de actuación y con múltiples variaciones en los escenarios que simulan un evento de crisis.

- Se han cumplido los objetivos y se puede afirmar que la arquitectura propuesta para entrenamiento conjunto real/virtual en gestión de emergencias ofrece una solución innovadora, interoperable, extensible, flexible y escalable que ayuda a mejorar el conocimiento en el área de gestión de emergencias y los procedimientos de actuación en eventos de crisis.

## **6.2. Desarrollos futuros**

A partir de los resultados obtenidos y de las observaciones hechas durante la realización de la tesis doctoral, se pueden identificar múltiples líneas de trabajo futuro:

- Sería interesante llevar a cabo una evaluación de calidad de experiencia con los usuarios del sistema. Esto, además de permitir identificar posibles fallos, proporcionaría información valiosa para mejorar la usabilidad del sistema.
- Otra posible línea de trabajo consistiría en implementar funcionalidades que ayuden a la gestión y configuración de sistemas de entrenamiento tales como: registro de errores,

definir/modificar condiciones meteorológicas, programar respuestas automáticas ante alertas del sistema, y/o sugerir alternativas de actuación ante amenazas.

- Podría contemplarse la implementación de protocolos como RTSP o RTP para gestionar los flujos de vídeo, así como la definición de parámetros de control relacionados con la adecuada gestión de un sistema de tiempo real.
- Otra opción interesante, facilitada por el uso de JSON para el intercambio de datos, sería crear una aplicación que tome todos los documentos XML generados con el servicio web y los reproduzca de manera continua, simulando un flujo de vídeo. De esta manera, si por algún motivo no es posible reproducir los flujos de vídeo almacenados en el servidor de vídeo virtual (p. e. limitaciones en recursos computacionales), se tendría otra alternativa para poder visualizar el entrenamiento simulado tras la realización del ejercicio y evaluar diferentes estrategias de actuación. Incluso podría implementarse en un entorno de realidad aumentada.
- La gestión de la coherencia es otra línea de investigación importante. El entrenador táctico implementa actualmente técnicas básicas de gestión de la coherencia y la consistencia pero la incorporación de técnicas avanzadas se prevé como línea de trabajo futuro en el grupo de investigación.
- También se tiene previsto en el grupo continuar explorando diferentes plataformas para gestión de mundos virtuales

(actualmente se está trabajando también con OpenSim), incluyendo la integración de sensores estacionarios y VANT (vehículo aéreo no tripulado), y desarrollar entornos virtuales que representen diferentes tipos de entornos reales. Otra línea de investigación de interés para futuros desarrollos es la sincronización en entornos distribuidos.

- Podrían incluirse nuevos formatos de descripción, basados en la normativa MPEG-V, para generar efectos sensoriales en entornos de realidad aumentada, (p. e. modificar las capacidades y comandos específicos de un sensor para controlar la cantidad de luz del ambiente).



## Glosario de términos

---

3D3C	<i>3D visualization &amp; navigation, Community, Creation and Commerce.</i> Acrónimo alternativo para referirse a los mundos virtuales.
API	<i>Application Programming Interface.</i> Conjunto de funciones y procedimientos que pueden utilizados por otro programa.
Avatar	Representación gráfica de un usuario, usualmente 2D o 3D, mediante la cual interactúa con un mundo virtual.
BIFS	<i>Binary Format for Scenes.</i> Basado en VRML y MPEG-4, es un formato binario para representar contenido audiovisual en 2D o 3D.
Broadcast	Mecanismo de transmisión (difusión) donde los paquetes se transmiten una sola vez y son escuchados por todos los nodos de la red en cuestión.
C2IS	<i>Command and Control Information System</i> - Sistema de información de mando y control. Aquel en el que existe una entidad responsable de gestionar los

---

---

	medios disponibles para cumplir un objetivo.
C4ISR	<i>Command Control, Computers and Communications Information Surveillance and Reconnaissance</i> - Sistemas de Mando, Comunicaciones, Control, Computadoras, Inteligencia, Vigilancia y Reconocimiento. Representan la evolución de los C2IS.
Conciencia de la situación	Comprensión compartida que se tiene de una situación. Implica el análisis y retroalimentación de lo que está sucediendo y es indispensable para la adecuada toma de decisiones. Suele utilizarse el término en inglés <i>situational awareness</i> .
Háptica	Se refiere a cualquier tipo de interacción que tenga que ver con el tacto.
ICS	<i>Incident Command System</i> - Sistema de comando de incidentes. Estructura organizativa para atender incidentes, reales o potenciales, con procedimientos basados en protocolos y que fomenta la comunicación efectiva entre todos los organismos participantes.
First responders	Aquellos que asisten en primer lugar cuando se produce una situación emergencia (usualmente paramédicos, bomberos, etc.).
MPEG-V	Estándar ISO/IEC 23005. Su objetivo es ofrecer una base técnica sólida para aplicaciones y servicios multimedia inmersivos y multidimensionales.

---

---

Middleware	Software intermedio que intercomunica varias aplicaciones y hace posible su funcionamiento sobre plataformas heterogéneas.
Multicast	Mecanismo de transmisión donde los paquetes se transmiten una vez y sólo son escuchados que pertenecen a la sesión multicast en curso, sin interferir con el resto de estaciones
RTSP	<i>Real Time Streaming Protocol</i> . Protocolo no orientado a conexión para establecer y controlar flujos sincronizados de datos.
Ruby on Rails	Infraestructura de aplicaciones web de código abierto, escrita en el lenguaje de programación Ruby.
SDK	<i>Software Development Kit</i> , kit de desarrollo de software. Conjunto de herramientas que permite desarrollar aplicaciones de software personalizadas.
Streaming	Mecanismo de transmisión de contenidos multimedia a través de una red en la que el usuario consume el contenido a medida que se descarga.
TIC	Tecnologías de la información y la comunicación.
TCP	<i>Transmission Control Protocol</i> . Protocolo de Control de Transmisión. Protocolo que proporciona el envío de datos seguro mediante reconocimientos y retransmisiones. Se encuentra documentado en la RFC 793.

---

---

URI *Uniform Resource Identifier.* Esquema de nombrado estándar para recursos web. Se encuentra documentado en la RFC 2396.

---

URL *Uniform Resource Locator.* Subconjunto de URIs que hacen referencia a ubicaciones específicas basadas en nombres de equipos o hosts.

---

VRML *Virtual Reality Modeling Language.* Conocido también como *Virtual Reality Markup Language*, lenguaje de marcado de realidad virtual, es el formato de archivos estándar para representar gráficos vectoriales 3D interactivos. Es predecesor de X3D.

---

X3D Sucesor de VRML, es un formato estándar de archivos basado en XML para representar gráficos computacionales en 3D.

---

XML *eXtensible Markup Language.* Lenguaje de marcado extensible. Se trata de un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

---

XSD *XML Schema Definition.* Esquema XML. Lenguaje para describir la estructura de un documento XML.

---

XSI *XML Streaming Instructions.* Atributo que define el espacio de nombres que va a ser utilizado en un esquema XML.

---

---

W3C	<i>World Wide Web Consortium.</i> Comunidad internacional que se dedica a desarrollar estándares web.
-----	---

---

Widget	Aplicación ligera, normalmente embebida en una página web, cuyo objetivo es dar fácil acceso a funciones frecuentemente usadas, con visualización dinámica e interactiva.
--------	---

---



# **Anexos**





# **Anexo A**

## **Publicaciones**

### **A.1. Congresos Internacionales**

- ARDILA, LAURA; PÉREZ-LLOPIS, ISRAEL; ESTEVE, MANUEL; y PALAU, CARLOS E.: LVC Training Environment for Strategic and Tactical Emergency Operations. En: 10th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2013). Baden-Baden, Germany, 2013 — ISBN 9783923704804, pp. 140-144.

### **A.2. Revistas indexadas**

- ARDILA, LAURA; PÉREZ-LLOPIS, ISRAEL; ESTEVE, MANUEL; y PALAU, CARLOS E.: Interoperable architecture for joint real/virtual training in emergency management using the MPEG-V standard. En: Computer Standards & Interfaces vol. 41 (2015), pp. 39-55.

# Anexo B

## Modelo de datos

### B.1. Modelo de datos MPEG-V genérico

```
<?xml version="1.0" encoding="utf-8"?>
<!--MPEG-V Genérico: Wrapper base para los elementos del
entrenador táctico-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:gen="GEN-NS" xmlns:vwoc="urn:mpeg:mpeg-v:2010:01-
VWOC-NS" xmlns:cidl="urn:mpeg:mpeg-v:2010:01-CIDL-NS"
xmlns:iidl="urn:mpeg:mpeg-v:2010:01-IIDL-NS"
targetNamespace="GEN-NS" id="MPEG-V-GEN-UPV.xsd">
  <xs:import namespace="urn:mpeg:mpeg-v:2010:01-
VWOC-NS" schemaLocation="MPEG-V\MPEG-V-VWOC.xsd"/>
  <xs:import namespace="urn:mpeg:mpeg-v:2010:01-
CIDL-NS" schemaLocation="MPEG-V\MPEG-V-CIDL.xsd"/>
  <xs:import namespace="urn:mpeg:mpeg-v:2010:01-
IIDL-NS" schemaLocation="MPEG-V\MPEG-V-IIDL.xsd"/>
  <!--Se define el nodo raíz y los hijos que
corresponden-->
  <xs:element name="Root">
    <xs:complexType>
      <xs:sequence>
        <!--Objetos Virtuales-->
        <xs:element name="VirtualObjects"
type="vwoc:VWOCInfoType" minOccurs="0"/>
        <!--Elementos del entrenador
táctico-->
        <xs:element name="SystemElements"
type="gen:SystemElementsType"/>
        <!--Elementos de control, si se
definen-->
```

```

                <xs:element name="Control"
type="cidl:ControlInfoType" minOccurs="0"/>
                <!--Parámetros de interacción, si
existen-->
                <xs:element name="Interaction"
type="iidl:InteractionInfoType" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <!--Definición elementos-->
    <xs:complexType name="SystemElementsType">
        <xs:sequence>
            <xs:element name="Alarm"
type="gen:AlarmComplexType" minOccurs="0"/>
            <xs:element name="Threat"
type="gen:ThreatComplexType" minOccurs="0"/>
            <xs:element name="Position"
type="gen:PositionComplexType" minOccurs="0"/>
            <xs:element name="Sensor"
type="gen:SensorComplexType" minOccurs="0"/>
            <xs:element name="Message"
type="gen:MessageComplexType" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <!--Alarmas-->
    <xs:complexType name="AlarmComplexType">
        <xs:sequence>
            <xs:element name="AlarmID"
type="xs:int"/>
            <xs:element name="AlarmType"
type="xs:int"/>
            <xs:element name="Text"
type="xs:string"/>
            <xs:element name="PosXYZ"
type="gen:PosXYZComplexType"/>
            <xs:element name="Origin"
type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <!--Amenazas-->
    <xs:complexType name="ThreatComplexType">
        <xs:sequence>
            <xs:element name="ThreatId"
type="xs:int"/>
            <xs:element name="ThreatType"
type="xs:string"/>

```



```

                permite definir comandos
específicos para cada Type de sensor-->
                </xs:sequence>
            </xs:complexType>
            <!--Mensaje-->
            <xs:complexType name="MessageComplexType">
                <xs:sequence>
                    <xs:element name="IdMessages"
type="xs:int"/>
                    <xs:element name="Destination"
type="xs:string"/>
                    <xs:element name="DateTime"
type="xs:date"/>
                    <xs:element name="Text"
type="xs:string"/>
                    <xs:element name="Origin"
type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="PosXYZComplexType">
                <xs:sequence>
                    <xs:element name="X" type="xs:long"/>
                    <xs:element name="Y" type="xs:long"/>
                    <xs:element name="Z" type="xs:long"/>
                </xs:sequence>
            </xs:complexType>
        </xs:schema>

```

## B.2. Modelo de datos MPEG-V para SIMACOP

```

<?xml version="1.0"?>
<!--MPEG-V-SIMACOP.xsd: Formato de datos de SIMACOP
adaptado a MPEG-V-->

<xs:schema targetNamespace="SMC-NS"      id="MPEG-V-
SIMACOP.xsd"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:gen="GEN-NS"
    xmlns:smc="SMC-NS"
    xmlns:vwoc="urn:mpeg:mpeg-v:2010:01-VWOC-NS"

```

```
xmlns:cidl="urn:mpeg:mpeg-v:2010:01-CIDL-NS" >

  <import namespace="urn:mpeg:mpeg-v:2010:01-VWOC-NS"
schemaLocation="MPEG-V\MPEG-V-VWOC.xsd"/>
  <import namespace="urn:mpeg:mpeg-v:2010:01-CIDL-NS"
schemaLocation="MPEG-V\MPEG-V-CIDL.xsd"/>
  <import namespace="GEN-NS" schemaLocation="MPEG-V-
GEN-UPV.xsd"/>

  <xs:element name="SimacopNode"
type="smc:SimacopType"/>

  <xs:complexType name="SimacopType">
    <xs:sequence>
      <xs:element name="Alarm"
type="gen:AlarmComplexType"/>
      <xs:element name="Threat"
type="gen:ThreatComplexType" minOccurs="0"/>
      <xs:element name="Object"
type="smc:ObjectType" minOccurs="0"/>
      <xs:element name="Position"
type="gen:PositionComplexType" minOccurs="0"/>
      <xs:element name="Sensor"
type="gen:SensorComplexType" minOccurs="0"/>
      <xs:element name="Message"
type="gen:MessageComplexType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <!--Tipo propio de SIMACOP-->
  <xs:complexType name="ObjectType">
    <xs:sequence>
      <xs:element name="ObjectId" type="xs:int" />
      <xs:element name="ObjectsText"
type="xs:string" />
      <xs:element name="Text" type="xs:string"/>
      <xs:element name="Origin" type="xs:string"/>
      <xs:element name="Position"
type="gen:PositionComplexType" />
      <!--Identification, description, content,
behavior-->
      <!--<xs:element name="vwoBase"
type="vwoc:VWOBaseType" minOccurs="0"/>-->
    </xs:sequence>
  </xs:complexType>
```

```
</xs:schema>
```

### B.3. Modelo de datos MPEG-V para OLIVE

```
<?xml version="1.0"?>
<!--MPEG-V-OLIVE.xsd: Formato de datos de OLIVE adaptado
a MPEG-V-->

<xs:schema targetNamespace="OLI-NS"      id="MPEG-V-
OLIVE.xsd"
      xmlns="http://www.w3.org/2001/XMLSchema"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:gen="GEN-NS"
      xmlns:oli="OLI-NS"
      xmlns:vwoc="urn:mpeg:mpeg-v:2010:01-VWOC-NS"
      xmlns:cidl="urn:mpeg:mpeg-v:2010:01-CIDL-NS"
    >

  <import namespace="urn:mpeg:mpeg-v:2010:01-VWOC-NS"
    schemaLocation="MPEG-V\MPEG-V-VWOC.xsd"/>
  <import namespace="urn:mpeg:mpeg-v:2010:01-CIDL-NS"
    schemaLocation="MPEG-V\MPEG-V-CIDL.xsd"/>
  <import namespace="GEN-NS" schemaLocation="MPEG-V-
    GEN-UPV.xsd"/>

  <xs:element name="OliveNode" type="oli:OliveType"/>

  <xs:complexType name="OliveType">
    <xs:sequence>
      <xs:element name="Avatar"
        type="oli:AvatarType" minOccurs="0"/>
      <xs:element name="VWO" type="oli:VWOType"
        minOccurs="0"/>
      <xs:element name="Inventory"
        type="oli:InventoryType" minOccurs="0"/>
      <xs:element name="Product"
        type="oli:ProductType"/>

      <!--xs:elementos sistema C4ISR-->
      <xs:element name="Alarm"
        type="gen:AlarmComplexType"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```
        <xs:element name="Threat"
type="gen:ThreatComplexType" minOccurs="0"/>
        <xs:element name="Object"
type="oli:ObjectType" minOccurs="0"/>
        <xs:element name="Position"
type="gen:PositionComplexType" minOccurs="0"/>
        <xs:element name="Sensor"
type="gen:SensorComplexType" minOccurs="0"/>
        <xs:element name="Message"
type="gen:MessageComplexType" minOccurs="0"/>

    </xs:sequence>
</xs:complexType>

<xs:complexType name="AvatarType">
    <xs:sequence>
        <xs:element name="UserID" type="xs:int"/>
        <xs:element name="AvatarName"
type="xs:string"/>
        <xs:element name="AvatarDoid"
type="xs:string"/>
        <xs:element name="Position"
type="gen:PositionComplexType"/>
        <xs:element name="TemplateId"
type="xs:int"/>
        <xs:element name="Status" type="xs:string"/>
        <!--Detalles de apariencia e información de
control. He conservado sólo algunas etiquetas para hacer
una caracterización básica de cada avatar-->
        <xs:element name="AvatarFullDesc"
type="vwoc:AvatarType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="VWOType">
    <xs:sequence>
        <xs:element name="ObjectId" type="xs:int"/>
        <xs:element name="Type" type="xs:string"/>
        <xs:element name="Owner" type="xs:int"/>
        <xs:element name="Attributes"
type="oli:AttributesType"/>
        <!--Detalles de apariencia e información de
control-->
```



```
        <xs:element name="VOFullDesc"
type="vwoc:VirtualObjectType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="InventoryType">
    <xs:sequence>
        <xs:element name="Pob" type="oli:PobType"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="AttributesType">
    <xs:sequence>
        <xs:element name="AttributeName"
type="xs:string" />
        <xs:element name="Type" type="xs:int"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="PobType">
    <xs:sequence>
        <xs:element name="Poid" type="xs:int"/>
        <xs:element name="Name" type="xs:string"/>
        <xs:element name="Type" type="xs:int"/>
        <xs:element name="RelatedPoid"
type="xs:int"/>
        <xs:element name="Owner" type="xs:int"/>
        <xs:element name="PurchaseTime"
type="xs:date"/>
        <xs:element name="PurchasePrice"
type="xs:decimal"/>
        <xs:element name="ProductsPid"
type="xs:string"/>
        <xs:element name="Phid" type="xs:string"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="ProductType">
    <xs:sequence>
        <xs:element name="ProductId" type="xs:int"/>
        <xs:element name="ProductType"
type="xs:string"/>
        <xs:element name="CreationDate"
type="xs:date"/>
    </xs:sequence>
</xs:complexType>
```

```

        <xs:element name="LastUpdateDate"
type="xs:date"/>
        <xs:element name="Description"
type="xs:string"/>
        <xs:element name="Name" type="xs:string"/>
        <xs:element name="Price" type="xs:decimal"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="ObjectType">
    <xs:sequence>
        <!--C4ISR-->
        <xs:element name="IdObject" type="xs:int" />
        <xs:element name="TextObjects"
type="xs:string" />
        <xs:element name="Text" type="xs:string"/>
        <xs:element name="Origin" type="xs:string"/>
        <xs:element name="Position"
type="gen:PositionComplexType" />

        <!--OLIVE-->
        <!--Types OLIVE: static, dynamic, usable
(interactable, carryable, products), rideable-->
        <xs:element name="Type" type="xs:string"/>

        <!--MPEG-V-->
        <!--Identification, description, content,
behavior-->
        <xs:element name="vwoBase"
type="vwoc:VWOBaseType" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

</xs:schema>
```