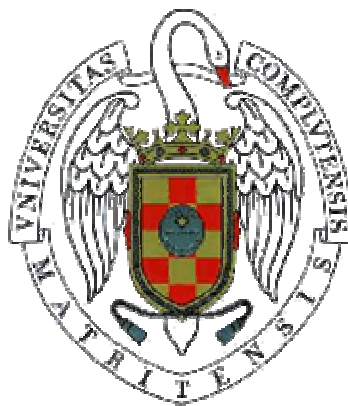


# Una Arquitectura General para la Integración de Videojuegos Educativos en Entornos Virtuales de Enseñanza



## **Proyecto de Fin de Máster en Sistemas Inteligentes**

Ángel del Blanco Aguado

*Directores:*

Baltasar Fernández-Manjón

Pablo Moreno-Ger

Facultad de Informática  
Universidad Complutense de Madrid  
Madrid, 2009



*El/la abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster:*

**“Una Arquitectura General para la Integración de Videojuegos Educativos en Entornos Virtuales de Enseñanza”**, *realizado durante el curso académico 2008-2009 bajo la dirección de Baltasar Fernández Manjón y Pablo Moreno Ger en el Departamento de ISIA, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.*

Madrid, 14 de septiembre de 2009

Ángel del Blanco Aguado



## Agradecimientos

En primer lugar quiero agradecer a mis directores de Proyecto de Final de Máster Baltasar Fernandez Manjón y Pablo Moreno Ger su apoyo y ayuda a la hora de realizar este trabajo. No hubiera sido posible llegar hasta el final sin su tutela.

Además quiero agradecer el apoyo recibido por parte de mis padres. Horten, Luis, gracias por todo. También se lo quiero agradecer a mi hermano Andrés y a Lola, por su paciencia y por sus ánimos. Sin olvidar al resto de miembros de mi familia, preocupándose por mí y mandándome aliento y apoyo desde la distancia.

Otra persona que me ha ayudado en muchos sentidos es mi compañero de “batallas” en la facultad de los últimos años. Gracias por tu ayuda, siempre con una sonrisa. Gracias Javi.

Por último y no por ello menos importante, a Gus, Dani, David, Ignacio, Ger, Joni, Iker, Didy, Carmen, Cris, a todos mis conocidos en la facultad y tantos otros que habéis escuchado cuando lo he necesitado. Sin olvidarme de Isa y de Vane, que en las últimas semanas han sabido sacarme una sonrisa. Y a “Lenny”, mi gato, que tanta compañía me ha hecho los largos días de escritura en casa. Gracias a todos.



# INDICE

<b>Resumen</b> .....	9
<b>Summary</b> .....	11
<b>Palabras clave:</b> .....	13
<b>Keywords:</b> .....	15
1. Introducción.....	19
1.1 Motivación.....	20
1.2 Objetivos.....	20
1.3 Estructura.....	21
2. Estudio del Dominio.....	23
2.1 Trabajo Relacionado.....	23
2.2 Estándares y Especificaciones en <i>e-Learning</i> .....	25
2.2.1. Descripción.....	25
2.2.2. Repaso histórico de los estándares en <i>e-Learning</i> .....	25
2.2.3 Revisión de Estándares y Especificaciones.....	26
2.2.3 SCORM.....	28
The SCORM Content Aggregation Model CAM.....	30
The SCORM Run-Time Environment RTE.....	35
The SCORM Sequencing and Navigation SN.....	43
2.3 Entornos Virtuales de Enseñanza (EVE).....	55
2.3.1 Descripción.....	56
2.4 Conclusiones del Capítulo.....	62
3. Arquitectura.....	63
3.1 Visión general de la arquitectura.....	63
3.2 El Modelo de Datos Abstractos.....	65
3.3 La Capa de Comunicación.....	66
3.4 La Capa de Adaptación y Evaluación.....	68
3.5 Conclusiones del Capítulo.....	68
4. Integración con <e-Adventure>.....	71
4.1 La plataforma <e-Adventure>.....	71
4.2 Implementación de la arquitectura en la plataforma <e-Adventure>.....	73
4.2.1 Implementación de la Capa de Adaptación y Evaluación.....	74
4.2.2 Implementación de la Capa de Comunicación.....	77
4.3 Conclusiones del Capítulo.....	82
5. Caso de Estudio.....	83
5.1 Presentación del caso de estudio.....	83
5.2 Detalle del caso de estudio.....	83
5.3 Configuración de las capas.....	85
6. Conclusiones y Trabajo Futuro.....	89
7. Referencias.....	93
8. Anexos.....	97
Anexo A. “ <i>Bridging the Gap: Adaptive Games and Student-Centered VLEs</i> ”.....	97
Anexo B. “ <i>A General Architecture for the Integration of Educational Videogames in Standards-compliant Virtual Learning Environments</i> ”.....	109



## Resumen

Hoy por hoy, la amplia utilización del *e-Learning* en empresas e instituciones educativas nos da una idea de la aceptación del uso de las nuevas tecnologías en educación, ya sea como un complemento a la formación (blended learning) o como parte principal del proceso de enseñanza. Este hecho ha fomentado que el progreso en las últimas décadas en el campo del *e-Learning* haya sido elevado tanto en contenidos como en la evolución de los sistemas que los soportan. Por ello son numerosos los esfuerzos por parte de diversas organizaciones para diseñar estándares que faciliten la interoperabilidad, reutilización y durabilidad de los contenidos.

Dentro del *e-Learning* encontramos una tendencia que fomenta el uso de videojuegos en educación. Parecen una buena herramienta para enseñar debido a que presentan determinadas características que favorecen la transmisión de conceptos, como aumentar la motivación. Además, al ser contenidos altamente interactivos permiten realizar evaluaciones del alumno más precisas, así como adaptar el desarrollo de los juegos teniendo en cuenta determinados factores (pasadas interacciones del alumno, conocimiento inicial, factores socioculturales, etc.).

A la hora de utilizar videojuegos en los modernos servidores educativos encontramos diversos problemas. Por un lado, la diversidad de estándares y la falta de una amplia aceptación de alguno de ellos. Esto supone una traba a la hora de determinar cuál usar para integrar estos contenidos en servidores educativos. Por otro, la diversidad de videojuegos educativos y el hecho de que los estándares actuales no están pensados para aprovechar al máximo las posibilidades adaptativas y de evaluación de contenidos altamente interactivos.

Parece interesante introducir en el proceso de creación de videojuegos educativos a los profesionales de la enseñanza para que puedan potenciar su valor pedagógico. Las dificultades tecnológicas que supone adoptar un estándar o adaptar y evaluar videojuegos alejan a los docentes de su participación en dicho proceso.

Este trabajo presenta una arquitectura multicapa que pretende afrontar los problemas derivados de la integración de videojuegos en el flujo educativo.



## Summary

Nowadays the wide spread use of *e-Learning* in educational institutions and companies gives us an idea of the wide acceptance of the use of new technologies in education not only as a complementary tool for education (blended learning) but also as part of the main educational process. This is the reason why the contents and their supporting systems in *e-Learning* have been developed to enhance the educational experience. That is why different organizations have made several efforts to design standards that make the interoperability, the reuse and durable contents easier.

We see inside *e-Learning* a clear tendency of using videogames as part of education. They seem to be a good tool to teach as they show certain characteristics that favours the transmission of contents such as motivational increase. Moreover as they show highly interactive contents they allow having more accurate evaluations of the students, as well as adapting the game flow taking into account specific patterns such as previous interactions with the student, initial knowledge, cultural and social factors...

The use of videogames in learning environments also presents different obstacles. On the one hand there seems to be a broad diversity of standards with also a broad lack of acceptance of the same which implies an obstacle to determine which of them should be used to integrate the contents in educational servers. On the other hand we consider the variety of educational videogames and the fact that actual standards are not capable to take maximum advantage of the adaptative possibilities and the assessment of highly interactive contents.

It seems interesting to take into consideration the opinion of people who are involved in education within the development of videogames in order to increase its pedagogical value. The technical difficulties that imply to adopt a standard, adapt and assess videogames seem to cast away teachers from this process.

This work presents a multilayer architecture that tackles issues that emerges from the integration of videogame in the process of education.



**Palabras clave:**

e-Learning

Videojuegos educativos

Entornos virtuales de educación

Aprendizaje adaptativo

Evaluación

SCORM



**Keywords:**

e-Learning

Standards

Educational videogames

Virtual learning environments

Adaptive Learning

Assessment

SCORM







## 1. Introducción

Debido a la multitud de avances tecnológicos y al rápido avance de Internet, se puede considerar que hoy por hoy vivimos en una sociedad en la que la comunicación y la información tienen un gran peso. Pero no debemos olvidar la importancia de la formación. Ya en el siglo IV antes de Cristo Aristóteles nos definió como “animales político-sociales”, y se puede decir que hay pocos procesos que sean más político-sociales que la enseñanza. Y como animales político-sociales que somos, necesitamos influir sobre el receptor. Esto se consigue por medio de la comunicación. La comunicación está en la base de todo proceso de enseñanza-aprendizaje.

Desde un punto de vista tecnológico, encontramos que las TIC (Tecnologías de la Información y la Comunicación) han cambiado la forma tradicional del proceso de enseñanza-aprendizaje desde el momento que se empiezan a utilizar ordenadores para tal fin, lo que se conoce como *e-Learning*. Encontramos el Computer Based Training (CBT) o *Computer-Assisted Instruction (CAI)* como una de las primeras aproximaciones del uso de ordenadores con fines educativos. Unido a esto, la utilización de Internet ha favorecido que el acceso a la información en cualquier lugar del mundo y en cualquier momento. Más en concreto, y para el tema que nos atañe, cualquier alumno puede desde cualquier lugar recibir materiales, realizar pruebas, enviar trabajos, o utilizar cualquier material educativo que sus profesores hayan alojado en la red global. Los materiales a los que puede acceder pueden ser contenidos formativos no interactivos, como apuntes, transparencias, etc., o contenidos formativos interactivos, con los que el alumno puede interactuar y validar su experiencia de aprendizaje (autoevaluaciones, ejercicios, juegos, etc...). Además facilita el proceso de comunicación tanto con su grupo de trabajo como con los profesores mediante la utilización de canales asíncronos (foros) y síncronos (chat), lo cual es de gran utilidad tanto para resolver dudas como para validar su aprendizaje.

El progreso en las últimas décadas en el campo del *e-Learning* ha sido elevado, ya sea desde el punto de vista tecnológico como desde el de los contenidos educativos que alberga. Los sistemas de *e-Learning* han evolucionado de ser repositorios de contenidos estáticos a sistemas más complejos en los que no solamente se almacenan contenidos, sino que se integran multitud de herramientas que cubren el proceso de aprendizaje. Son conocidos como Learning Management Systems (LMS), en español Sistemas de Administración del Aprendizaje, o Virtual Learning Environments, en español Entornos Virtuales de Enseñanza (EVE). Con el uso de estos sistemas se intenta que el alumno tome un rol activo en la enseñanza. Además se fomenta el aprendizaje colaborativo mediante el uso de herramientas de grupo. Los EVE permiten utilizar contenidos interactivos y almacenar las interacciones de cada alumno para realizar evaluaciones de su actividad. Otra característica interesante es que estos sistemas se basan en entornos Web, lo que permite un fácil acceso a los cursos virtuales.

En relación con la variedad de contenidos que se pueden utilizar para transmitir información y conocimientos, encontramos los juegos como un complemento a la formación. A la par que se empezaban a utilizar CBT en determinadas empresas e instituciones, autores como (Malone 1981) empiezan a valorar las posibilidades educativas derivadas de utilizar videojuegos e identificaron diversas características de los juegos que son interesantes para transmitir conceptos por su potencial didáctico (Malone y Lepper 1987), destacando principalmente que aumentan la motivación de los alumnos. Esto fomentó la discusión académica de las posibilidades educativas de los juegos (Leutner 1993;

Papert 1998; Prensky 2001; Gee 2003; Michael and Chen 2006; D. Michael and Chen 2006; Aldrich 2005) y su posible introducción en el flujo educativo. Aunque no existe una aceptación y utilización lo suficientemente amplia, hoy en día encontramos multitud de tipos de videojuegos destinados a ser utilizados en educación (Tang 2007). La discusión se ha trasladado hacia otro tipo de aspectos, tales como su integración en EVE (Torrente et al 2009) y las posibilidades de seguimiento de la actividad del alumno y adaptación del contenido (Moreno-Ger et al 2007a; de Freitas and Oliver 2006).

## 1.1 Motivación

Al mismo tiempo que se han ido desarrollando los EVE y han ido incrementando la complejidad de los contenidos que soportan, se han ido desarrollando protocolos y estándares de *e-Learning* para facilitar la interoperabilidad de los contenidos y que puedan utilizados en distintos sistemas. La diversidad de estándares unido a que ninguno de ellos está ampliamente aceptado supone un problema que dificulta uno de sus propósitos: la interoperabilidad de los contenidos.

Este problema se acentúa cuando los contenidos a integrar en EVE son videojuegos debido a su alta interactividad. Esta característica permite realizar una evaluación más refinada de la interacción del alumno, de igual manera que permite utilizar la información recogida en estas evaluaciones para adaptar este tipo de contenidos. Los estándares están pensados para integrar contenidos menos interactivos (hipermedia, HTML, etc) por lo que supone una dificultad añadida realizar tareas de adaptación y evaluación sobre los juegos.

Para facilitar la inclusión de videojuegos educativos en EVE se necesita alcanzar una gran independencia entre la implementación de los juegos y los estándares utilizados para conectarlos con los EVE para adaptación y evaluación

## 1.2 Objetivos

Una vez planteados los problemas, identificamos el siguiente conjunto de objetivos:

- 1) Diseñar una arquitectura que se presente como una capa que abstraiga la comunicación con los servidores educativos a los desarrolladores de juegos, de manera que puedan crear juegos adaptativos y que evalúen el rendimiento del alumno sin la necesidad de comprometerse con ningún estándar en concreto, y que puedan reutilizar el mismo juego para distintos EVE.
- 2) Proveer a la arquitectura de un mecanismo que permita definir determinadas situaciones relevantes de adaptación y evaluación sin que sea necesario tener un conocimiento exhaustivo ni del juego ni del servidor educativo en el que va a ser integrado.
- 3) Poner en práctica las ideas de la arquitectura en una plataforma de juegos concreta para comprobar su posible aplicación en entornos reales.

Cabe destacar que el objetivo principal de este trabajo es plantear una arquitectura que tenga en cuenta estos aspectos, centrándose principalmente en la comunicación entre EVE y juegos. Aunque son planteados quedan fuera del marco de este trabajo aspectos tales como cuales son las situaciones a evaluar o cómo se debe modificar el flujo del juego, así como la identificación de términos generales que expresen estos propósitos. Se tienen en cuenta y se dan soporte en la arquitectura para su refinamiento posterior.

### 1.3 Estructura

En este documento se realiza un estudio del dominio para posteriormente plantear la propuesta teórica de la arquitectura, su implementación en una plataforma de juegos concreta y se exponer un caso de estudio concreto en el que se ponga a prueba las ideas expuestas. Finalmente se dan unas conclusiones y posible trabajo futuro.

En la sección 2 se analiza el trabajo relacionado. Además se hace una revisión de los estándares en *e-Learning*, poniendo especial atención en el perfil de aplicación SCORM.

En la sección 3 se realiza la descripción teórica de la arquitectura. En ella se analiza con detalle cada uno de los elementos que la conforman.

En la sección 4 se expone la aplicación de las ideas expuestas en la sección anterior en una plataforma de juegos concreta. En ella se detallan los cambios y funcionalidades añadidas para introducir cada uno de los elementos de la arquitectura.

En la sección 5 se muestra un caso de estudio. Para ello se utiliza un juego creado con la plataforma en la que se ha añadido la arquitectura para integrarlo en diversos servidores educativos y comprobar que se comunican tal y como se definió.

Por último, en la sección 6 se muestran las conclusiones obtenidas y se da una descripción del posible trabajo futuro.



## 2. Estudio del Dominio

La integración de contenidos educativos en distintos sistemas no es nueva y esta ligada a la expansión de la utilización del Computer-Based Training tanto en la industria como en instituciones educativas. Fue motivado por los elevados costes que suponía que cada empresa desarrolladora de contenidos tuviera que proveer un hardware específico para ejecutarlos. Estos hechos provocaron que diversas organizaciones comenzaran a trabajar para encontrar un patrón o guía que unificará la distribución de contenidos y la forma de ejecutarlos en la mayoría gama posible de sistemas. Es decir, motivó la creación de estándares en *e-Learning* como base para la expansión de su uso.

Hoy en día encontramos multitud de estándares y especificaciones que abogan por la interoperabilidad de los contenidos educativos. Tal es el volumen y la ausencia de uno ampliamente aceptado que ha llegado a convertirse en un problema a la hora de integrar material educativo en los Entornos Virtuales de Enseñanza. También existen iniciativas y estudios para la utilización de estos estándares desde un punto de vista de adaptación y evaluación en contenido, y más en la dirección que aborda este trabajo, en tipos concretos de contenido, como las simulaciones y los juegos.

En este capítulo vamos a abordar el uso de estándares para la integración de contenidos educativos en EVE, y más concretamente en juegos. Primero se muestra trabajo relacionado en esta línea. Después se analizan los estándares con más impacto y más utilizados en *e-Learning* utilizando el trabajo (Fernández-Manjón et al 2007, Fallon y Brown 2003) de para pasar a describir con mayor nivel de detalle el perfil de aplicación SCORM. Finalmente, se realiza un estudio de algunos de los EVE actuales.

### 2.1 Trabajo Relacionado

Muchos son los esfuerzos actuales en la literatura para proponer ideas de cómo adaptar, evaluar e integrar en EVE determinados contenidos educativos. Los contenidos hipermedia han sido ampliamente utilizados en educación (Dillon and Gabbard 1998, Hutchings et al 1992). La diversidad de información mostrada en este tipo de contenidos ha favorecido su aceptación para crear contenidos educativos. Encontramos la definición de enseñanza asistida por hipermedia<sup>1</sup> en el trabajo de (Heller 1990) en el que se identifican distintos factores que potencian el aprendizaje utilizando este tipo de contenidos. En esta misma línea, el trabajo de (McKnight 1996) hace un análisis del campo. Ambos identifican como una de sus principales ventajas educativas que no son contenidos lineales, y (McKnight 1996) enfatiza en que a la hora de crear este tipo de contenidos hay que tener en cuenta determinados factores, como los usuarios que lo van a utilizar (en función de sus conocimientos iniciales, destrezas, hábitos, etc.), las tareas para las que va a ser utilizado y el tipo de información a mostrar (texto, imágenes, vídeo, etc.). Estos conceptos son la base de la adaptación en contenidos hipermedia. En el trabajo de (Brusilovsky 2001) se realiza un estado del arte de la adaptación en este tipo de contenidos en educación y en otros campos, como en el de la recuperación de información. Destaca dos factores fundamentales a la hora de adecuar los contenidos a los usuarios que los utilizan: el modelo de usuario y el diseño de hipermedia. Con el modelo de usuario se pretenden identificar las características en función de las cuales se modificará el desarrollo del contenido, como rasgos de personalidad, conocimiento, preferencias, etc. En cuanto el diseño de contenidos, distingue

---

<sup>1</sup> Hypermedia Assisted Instruction (HAI)

entre la adaptación del nivel de dificultad y adaptación en la secuencia de material, en la que se mostrará una u otra información en función del modelo de usuario. Para llevar a cabo estas tareas de adaptación, aparece el concepto de Sistema de Adaptación de Hipermedia<sup>2</sup>, como un servidor encargado de almacenar el modelo de usuario (pero no tiene porque almacenar los datos para cada usuario), almacenar contenido hipermedia y realizar adaptación del contenido en función de este modelo y los datos específicos de un usuario. Además, también ha sido fuente de estudio de que manera evaluar la actividad del alumno en este tipo de contenidos (Kommers y Lanzing 1997, Yamada et al 1995, Ng 2002).

En cuanto a la integración de contenidos hipermedia en EVE, encontramos diversas propuestas para facilitar las tareas de adaptación y evaluación de los mismos. En cuanto adaptación, (Nussbaumer et al 2007) propone una arquitectura para integrar contenidos hipermedia en EVE sin características adaptativas. Cabe destacar el módulo encargado de llevar a cabo la adaptación, el cual recibe información del EVE de las relaciones entre los contenidos que almacena y los datos almacenados para el usuario. Procesa esta información y en función del resultado muestra al alumno determinado contenido. En esta misma línea, (Conlan et al 2002) propone una arquitectura que intercomunique un EVE y un AHS. De esta manera se introducen las tareas adaptativas que realizan estos sistemas para modificar como se muestran al usuario los contenidos en función de los datos de usuario almacenados en el EVE. Es interesante que esta propuesta utilicen diversos estándares con diversos propósitos, como IMS Content Packaging para almacenar los contenidos, el modelo de datos SCORM para compartir datos de usuario entre ambos sistemas, o SCORM Runtime Communication API para realizar la comunicación.

Adentrémonos en la utilización de SCORM para integrar distintos contenidos. Encontramos una revisión de las ventajas e inconvenientes de utilizar el estándar en el campo de *e-Learning* en (Witthaus 2009). En el trabajo de (de Antonio et al 2008) presentan una extensión del secuenciamiento de SCORM para introducir la adaptación en simulaciones para conectar este tipo de contenidos con EVE y que se pueda modificar la manera en la que son mostrados. Además proponen una arquitectura para la comunicación entre las simulaciones y los servidores educativos. Otra propuesta similar es la de (Manikonda et al 2004), en la que también proponen una arquitectura que conecta simulaciones con EVE a través de este estándar, pero sin modificar ninguna de sus características. La arquitectura también facilita las tareas de evaluación de la actividad del alumno en este tipo de contenidos.

El trabajo de (Rey-López et al 2008) muestra una ampliación de SCORM que abarca bastantes mas campos del estándar que las propuestas anteriores. Se utiliza para integrarlo dentro de un sistema de tutoría inteligente que presenta contenidos que combinan programas de televisión y otros materiales para adaptar cómo son mostrados en función de diversos aspectos del perfil del estudiante. Cabe destacar que aunque realizan multitud de cambios en SCORM, proveen una herramienta de autoría que crea paquetes de contenido cien por cien compatibles con el estándar y que pueden ser integrados en cualquier servidor compatible con SCORM. Desde el punto de vista de la integración de juegos en EVE con SCORM, (Guin et al 2008) muestra cómo integrar videojuegos y evaluarlos usando este estándar. En el trabajo de (Chen et al 2009) se propone una idea bastante innovadora: un entorno de aprendizaje basado en video-juegos<sup>3</sup> en el que se pueden desarrollar y ejecutar contenidos educativos basados en juegos. Este sistema se comunica con otros EVE para

---

<sup>2</sup> Adaptative Hypermedia System (AHS)

<sup>3</sup> Video Game-Based Learning Environment

reutilizar contenidos que almacenen y obtener/enviar información de la actividad del usuario.

En cuanto a la integración de videojuegos en EVE encontramos los trabajos de (Torrente et al 2009, Moreno-Ger et al 2009). En el primero se hace una revisión de los estándares en *e-Learning* y se plantea como integrar videojuegos educativos en EVE usando dichos estándares para explotar sus características de adaptación y evaluación. Presenta un caso de estudio en el que se integra un videojuego en distintos EVE para probar el soporte de los mismos a determinadas tareas, tales como la evaluación de la actividad del alumno. En el segundo se realiza una revisión y se dan nuevas ideas de la utilización conjunta de juegos y EVE como parte del proceso de aprendizaje, abordando la adaptación y evaluación de este tipo de contenidos, y como combinar ambos elementos para aunar sus beneficios.

## **2.2 Estándares y Especificaciones en *e-Learning***

### **2.2.1. Descripción**

El BSI Group<sup>4</sup> define un estándar desde un punto de vista técnico como “un documento publicado que contiene una especificación técnica u otros criterios precisos para ser utilizados sistemáticamente como norma, directriz o definición”. Desde el punto de vista de producto, un estándar sirve como medio de entendimiento entre las distintas personas y/u organismos que intervienen en el proceso de creación y consumo de un producto.

Más concretamente en *e-Learning* el uso de estándares está estrechamente ligado a los contenidos. Determinadas maneras “estándar” de generar los contenidos se permite que estos se reutilicen, se intercambien entre distintos sistemas, perduren en el tiempo, se puedan gestionar, acceder a ellos, etc.

### **2.2.2. Repaso histórico de los estándares en *e-Learning***

Hagamos un repaso de histórico de los estándares en *e-Learning* para conocer su evolución y mejoras y explorar la relación entre ellos, ya que los estándares que hoy en día encontramos toman como punto de partida ideas y trabajos anteriores.

Aunque los primeros sistemas dedicados al entrenamiento asistido por ordenador (del inglés, Computer-Based Training, o CBT) datan de finales de los años 60, no es hasta la década de los 80 cuando se empieza a utilizar en organizaciones. La industria de la aviación fue una de las primeras industrias que empezó a utilizar el CBT para mantener la formación de sus empleados ante el desarrollo constante de los aviones. Decidieron apostar por este tipo de formación ya que encontraron interesante la posibilidad de utilizar contenidos más interactivos en los que se permitía de llevar un control del progreso y evaluación de los usuarios y a los que era fácil acceder por estar en red, invirtiendo millones de dólares en la producción de CBT. Tras unos años utilizando este tipo de sistemas dieron con el problema del excesivo coste de los contenidos. Estos se ejecutaban en hardware específico para cada compañía fabricante de aviones, la cual se encargaba de generar los cursos y suministrar las máquinas para ejecutarlos.

El Aviation Industry CBT Committee, más conocido como AICC, surge con el ánimo de solucionar este problema. Estaba formado por miembros de todos los organismos

---

<sup>4</sup> British Standards Institution

implicados en el proceso de utilización de CBT en la aviación: entrenadores de pilotos, fabricantes de aviones, desarrolladores de cursos de *e-Learning*, gobierno, etc. Una de sus primeras iniciativas fue la de estandarizar el hardware en el que iban a ejecutarse los cursos. Mas adelante apostaron por la interoperabilidad de los contenidos para que pudieran ser compartidos por todos los fabricantes en lo que se convirtió en las directrices más conocidas de AICC: la especificación CMI<sup>5</sup>. Utilizaba una red local para distribuir los contenidos entre las distintas empresas. Esta especificación se amoldó a la aparición de la formación basada en web (o WBT, del inglés Web Based Training) convirtiéndose en la primera especificación en pos de la interoperabilidad de contenidos en la red.

Los esfuerzos de estandarización realizados por AICC no servían simplemente para la industria de la aviación sino que podían ser aplicados en *e-Learning* en general. En particular, la especificación CMI de AICC se aplicó para compartir cursos de CBT y WBT siendo el primer estándar de *e-Learning* con una aceptación relevante.

A raíz de estos avances otras organizaciones comenzaron a trabajar en esta línea. IMS Project surgió como un consorcio sin ánimo de lucro de profesionales de la enseñanza fundamentalmente de universidades americanas, empresas editoriales, usuarios y empresas relacionadas con tecnologías de aprendizaje que produjeron un conjunto de especificaciones para estandarizar multitud de aspectos de *e-Learning*: metadatos, contenidos, etc. IMS pasó a ser una organización en la que contribuyen instituciones de todo el mundo, conociéndose desde ese momento como IMS Global Learning Consortium. La asociación IEEE no quiso quedarse atrás en este campo y colegió al LTSC<sup>6</sup> para el desarrollo de estándares educativos.

Del ministerio de defensa americano surgió en la iniciativa ADL<sup>7</sup> cuyo cometido era el de facilitar el acceso a la formación y el entrenamiento para alcanzar una mejor preparación de los militares de cara a un conflicto bélico. No solo surge del esfuerzo del ministerio de defensa americano, sino que en su creación nace de la colaboración de industrias privadas, organismos educativos o del gobierno federal de los Estados Unidos, entre otros, por lo que se ven satisfechas necesidades de todas estas instituciones. Fruto de la cooperación de estas organizaciones surge SCORM<sup>8</sup>. Es un perfil de aplicación que une los esfuerzos de estandarización realizados por AICC, IMS y LTSC mas aceptados hasta el momento. Aunque surge con fines de formación militar, es el modelo de referencia que mayor aceptación tiene por el momento, y es utilizado por multitud de organizaciones públicas y privadas.

### **2.2.3 Revisión de Estándares y Especificaciones**

#### Comité de CBT de la Industria de la Aviación (AICC)

La AICC desarrolló multitud de directrices técnicas tanto para hardware como para software para la estandarización del CBT y WBT. La más importante entre ellas por su repercusión y aceptación es la conocida como “especificación CMI”, que es un conjunto de pautas para la interoperabilidad entre cursos web y servidores EVE. Otra contribución a destacar es la del Modelo de Datos CMI, que sirve como estructura de la información relativa a la interacción del alumno con el contenido y como almacenarlo en un EVE.

---

<sup>5</sup> Computer Managed Instruction

<sup>6</sup> Learning Technology Standards Committee

<sup>7</sup> Advanced Distributed Learning

<sup>8</sup> Shareable Content Object Reference Model

Además ofrecen un test de certificación para distintos productos *e-Learning*. Con este test se puede verificar que las unidades asignables (AU, parecido al concepto actual de Objeto de Aprendizaje), los cursos y los LMS, entre otros productos, cumplan las guías de interoperabilidad CMI. Cabe destacar que el cumplimiento de este test no implica que se ajuste a las directrices de AICC.

### IMS Global Learning Consortium

IMS es el principal impulsor de especificaciones abiertas para distintos aspectos del *e-Learning* con el fin de conseguir la interoperabilidad de los contenidos. Entre sus especificaciones encontramos abordados aspectos que van desde el la creación de contenidos (diseño de actividades de enseñanza, etiquetado, empaquetamiento, secuenciación de actividades, etc.) a la evaluación (registro, informe de la actividad del alumno, intercambio de evaluaciones entre sistemas, etc.). Repasemos sus especificaciones más relevantes:

- **IMS Learning Resources Meta-Data:** definen una forma de describir determinada información adicional asociada al contenido educativo para conocer el objetivo que se busca con su utilización en la enseñanza. Con ello se busca mejorar su clasificación y recuperación en repositorios de contenidos.
- **IMS Content Packaging:** define como empaquetar contenidos educativos reutilizables. Su principal objetivo es que se puedan ejecutar en distintos EVE.
- **IMS Question & Test Interoperability:** define un modelo de representación de preguntas y como evaluarlas, de manera que dicha información pueda ser intercambiada entre EVE.
- **IMS Simple Sequencing:** define mecanismos que permitan definir el orden en que serán mostrados distintos contenidos educativos, pudiendo condicionar su ejecución en función de la actividad del alumno.
- **IMS Learning Design:** da soporte a distintos tipos de pedagogías. Mediante la especificación de un lenguaje permite la creación de metodología pedagógica que pueden reutilizarse para la creación de distintas sesiones de aprendizaje.

### IEEE Learning Technology Standards Committee

El IEEE/LTSC esta formado por distintos grupos de trabajo que desarrollan estándares técnicos y realizan guías para el desarrollo de software, tecnologías y métodos de diseño en el ámbito del *e-Learning*. Abarcan multitud de áreas de las tecnologías de la información para aprendizaje y educación. Se centran principalmente en el desarrollo, despliegue, mantenimiento e interoperabilidad de los componentes y sistemas *e-Learning*.

La especificación IEEE/LTSC LOM es el estándar de *e-Learning* que ha conseguido una adaptación mayor y surge como resultado del trabajo realizado por IMS y otro grupo conocido como ARIADNE. Toma muchas de las ideas de IMS Learning Resources Metadata y en ella se fija como se deben complementar con meta datos los OA.

## Alliance of Remote Instructional Authoring & Distribution Networks for Europe (ARIADNE)

A raíz de dos proyectos de investigación europeos llevados a cabo bajo este nombre surge la fundación ARIADNE, la cual promueve el desarrollo de herramientas y metodologías para producir, administrar y reutilizar contenidos educativos (OA).

### La iniciativa de ADL

ADL define un modelo de referencia de objetos de contenido compartido (lo que vienen a ser las siglas SCORM en inglés) el cual consiste en un perfil de aplicación que congrega algunas de las especificaciones y estándares vistos con anterioridad para intentar solventar los siguientes requerimientos en el aprendizaje basado en Web, tanto por los contenidos como por los servidores EVE: reutilización, accesibilidad, durabilidad e interoperabilidad.

Encontramos dos versiones que han tenido gran aceptación: SCORM v1.2 y SCORM 2004. En la primera de ellas encontramos tres documentos que hilan la propuesta de ADL:

- **The SCORM Overview:** da una visión general de la propuesta y de la estructura de SCORM.
- **The SCORM Content Agregation Model (CAM):** define como los contenidos educativos pueden ser identificados y descritos, y como se estructura en función de los recursos con contenga, es decir, como “empaqueta” los contenidos.
- **The SCOM Run-Time Environment (RTE):** define como lanzar y realizar la comunicación almacenando la actividad del estudiante en un entorno Web.

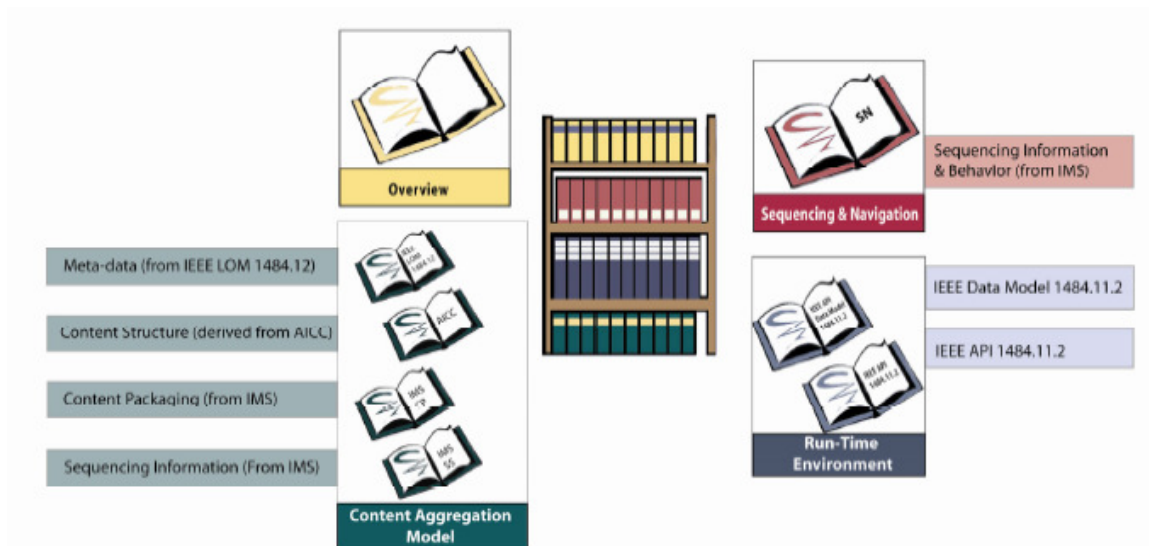
Para SCORM 2004, además de actualizar estos tres documentos con las mejoras y cambios incluidos en la versión, añade un nuevo documento en el que se amplían los conceptos que en la versión 1.2 se daban muy por encima en el libro CAM:

- **The SCORM Sequencing and Navigation (SN):** define como deberá desarrollarse la experiencia educativa, es decir, cual será el flujo y qué recursos se irán mostrando en función de las acciones realizadas por el estudiante.

Además para ambas versiones se provee de una test-suite para comprobar que tanto los contenidos como los EVE se amoldan a las recomendaciones dadas por SCORM.

### **2.2.3 SCORM**

SCORM surge de la iniciativa de ADL como un modelo de referencia creado tomando como punto de partida algunas de las aproximaciones tecnológicas presentadas en el apartado anterior. En el documento “The SCORM Overview” definen un modelo de referencia como “aquello que muestra que tipos de servicios son necesarios para resolver un problema en concreto, como pueden ser reunidos los estándares que utiliza y como deben ser usados”. El problema que pretende abordar es el de dar fundamentos tecnológicos al aprendizaje asistido por Web por medio de la utilización de las tecnologías emergentes. SCORM promueve el uso de la Web (aunque no sea idea original suya) como medio para distribuir los materiales educativos. Así pues se puede decir que SCORM no es un estándar o una especificación en sí, sino que se presenta como una “patrón a seguir” para llevar a cabo una meta determinada teniendo como referencia los trabajos realizados en el campo por otras organizaciones.



**Figura 1** Visión general de los estándares y especificaciones que engloba SCORM en su propuesta. Esta imagen ha sido obtenida del libro “The SCORM Overview”.

Se pretende que las guías que brinda SCORM sean utilizadas e implementadas por los creadores de contenidos educativos, de EVE, de herramientas que ayuden en ambos procesos y por los usuarios de las mismas.

SCORM toma las propiedades deseadas para los objetos educativos como los requerimientos de su modelo de aplicación con ánimo de fomentar su utilización. Estos son los requerimientos (ADL 2006):

- **Accesibilidad:** capacidad de encontrar, acceder y distribuir contenidos educativos alrededor del mundo.
- **Adaptabilidad:** capacidad de adecuar las instrucciones a las necesidades específicas de los usuarios.
- **Asequible:** capacidad de mejorar la eficacia y productividad disminuyendo los costes y el tiempo empleado para crear y distribuir materiales educativos.
- **Durabilidad:** capacidad de resistir los cambios tecnológicos sin un elevado coste.
- **Interoperabilidad:** capacidad de utilizar los materiales con herramientas o plataformas distintas a las que se utilizaron para crearlos.
- **Reutilizable:** capacidad de utilizar un mismo contenido en diferentes cursos y contextos.

También define requerimientos para los servicios que deben proveer los EVE basados en Web a la hora de distribuir y ejecutar los materiales didácticos:

- Ejecutar los contenidos creados por distintas herramientas de autoría e intercambiar datos con ellos.
- Que puedan acceder y lanzar contenidos almacenados en repositorios globales.

SCORM nos presenta una colección de libros técnicos en los cuales encontramos reunidos los estándares y especificaciones que conforman el modelo de referencia que propone. SCORM adopta en gran medida estas especificaciones y estándares y en determinadas situaciones complementa o modifica para aumentar su significado o funcionalidad.

## **The SCORM Content Aggregation Model CAM**

El objetivo del libro Modelo SCORM de Agregación de Contenidos es el de describir los compromisos y requerimientos a los que se deben amoldar cualquier organización que desee crear contenidos educativos, por medio de fijar qué componentes forman el contenido educativo, cómo empaquetarlos para que puedan ser distribuidos y ejecutados en EVE para compartir información con él, cómo etiquetarlos con la información necesaria (metadatos) para que sean almacenados y recuperados en los repositorios de contenidos, y cómo definir la información necesaria para secuenciar la ejecución de los componentes que los forman. El propósito general este libro es “fomentar la consistencia de almacenamiento, etiquetado, empaquetamiento, intercambio y descubrimiento de contenidos educativos”. En definitiva, representa un modelo de agregación de contenidos dirigido a los desarrolladores de contenidos con la finalidad de crear experiencias de aprendizaje de una manera imparcial y pedagógica.

Una de las claves en la creación de experiencias educativas es la recopilación de materiales educativos dándoles una estructura y organización, además de un sistema de catalogación y orden de ejecución de los materiales, pudiendo ser este lineal o dependiente de determinadas condiciones. Por tanto, las áreas en las que se centra el documento CAM son el modelo y empaquetamiento del contenido, los metadatos que llevan asociados tanto el paquete en general como los componentes en particular, y las “reglas” de secuencia y navegación que guiarán el desarrollo de la experiencia educativa.

### **Componentes del Modelo SCORM de Contenidos**

El modelo de contenido SCORM define cuales serán los componentes que forman los paquetes de contenidos, y cual será su organización. Encontramos las definiciones de Recurso (Asset), objetos de contenido compartibles (SCO), Actividad, Organización del Contenido, y Agrupación de los Contenidos.

#### **Recurso (Asset)**

Un recurso es cualquier contenido electrónico que pueda ser mostrado en una página web. Encontramos entonces que son recursos los textos, sonidos, imágenes, objetos flash, páginas web, entre otros. Se considera como la unidad básica dentro de la experiencia educativa. Además pueden ser catalogados por medio de metadatos para que puedan ser recuperados de un repositorio de contenido con fines de reutilización o mantenimiento.

#### **Objetos de Contenido Compartible (SCO)**

Los SCO se encuentran un peldaño por encima en el nivel de descripción de los componentes del modelo SCORM. Engloba un conjunto de Assets (como mínimo uno) y es la mínima unidad dentro de la experiencia educativa que puede comunicarse con el EVE para registrar su actividad y compartir información por medio del Modelo de Datos de SCORM RTE. La comunicación entre el SCO y el EVE se realiza vía SCORM RTE usando la API de IEEE ECMAScript. Este proceso queda descrito con mayor detalle en la sección de este documento que analiza el citado libro técnico.

Los SCO pueden ser englobados dentro de unidades más grandes, lo que se denominan Actividades, pero interesa la modularidad de las mismas con propósito de que cada SCO pueda ser reutilizado en distintos contextos. Para favorecer la reutilización ADL

recomienda que los SCO sean en lo posible independientes del contexto, encargándose las Actividades de aunar los SCO con un fin específico. Un concepto importante asociado a la creación de SCO siguiendo estas guías es su reutilización en diversos entornos sin importar qué herramienta se utilizó para su desarrollo o la Actividad en la que se encuentre.

Al igual que los Assets, los SCO pueden llevar metadatos asociados para su integración y recuperación en los repositorios.

### Actividades

Los SCO se presentan como unidades, preferentemente independientes del contexto en pos de su reutilización, capaces de comunicarse con el EVE para almacenar y evaluar la actividad del alumno. Las Actividades se proponen como parte de la estructuración de la experiencia educativa para dotar de contexto a los SCO y Asset, pudiendo ser definidas como “unidades con significado”. Las Actividades se conocen también con el término Ítem. Las Actividades pueden anidarse cuanto se considere necesario, es decir, una Actividad dentro de la Organización del Contenido puede ser a su vez una Actividad. En este caso las Actividades se conocen como agregación (CAM), o como cluster (SN). Así se permite amoldar las experiencias educativas a taxonomías de aprendizaje específicas al conceder la elección de la estructura completa al creador del contenido. Las Actividades hoja deberán llevar asociado un SCO o Asset, es decir, dotan de contenido el concepto estructural. Si vemos la estructura de una Organización como un árbol, las Actividades son todos y cada uno de los nodos, pudiendo tener un significado meramente estructural (nodos) o teniendo determinado material educativo asociado (hojas).

Se puede concluir que las Actividades son elementos meramente estructurales, y necesitan utilizar los recursos educativos englobados en SCO y Assets para dotar de “actividad” a la estructura.

Al igual que los elementos anteriores, las Actividades pueden llevar metadatos asociados, tanto para su catalogación en repositorios como para facilitar las tareas de mantenimiento.

### Organización del Contenido

La Organización del Contenido es el mapa o árbol que indica como es la estructura de la experiencia educativa. Esta formado por un conjunto de Actividades, que son los elementos atómicos de la organización. Muestra de una manera conceptual la interrelación de los elementos con significado que presenta la experiencia educativa que se va a crear.

En la Organización se describe como será la secuencia de las actividades mediante la estructura jerárquica que presenta y/o añadiendo información específica de secuencia y navegación a Actividades en concreto. En la primera, la secuencia viene dada por la misma estructura. En la segunda, se utilizan las indicaciones del libro técnico The SCORM Sequencing and Navigation para que el EVE lo interprete y elija posibles alternativas en el desarrollo la experiencia educativa mostrando unas u otras actividades en función de los progresos del alumno.

## Agregación del Contenido

La Agregación de Contenido da una idea de cómo definir entidades conceptuales con un determinado significado o fin y como interconectar entidades con funcionalidades similares para abarcar un objetivo mayor. La Agregación de Contenidos permite la reutilización de contenido. Además sirve para transmitir entre distintos sistemas el contenido y su estructura.

## **Empaquetamiento de los Contenidos**

Además de definir como será la estructura de los contenidos y qué componentes los forman, aparece la necesidad de definir una forma estándar de empaquetarlos para que puedan ser compartidos por distintos sistemas y herramientas. SCORM sigue la Especificación IMS para el Empaquetamiento de Contenidos (IMS Global Consortium. 2004) amoldándola ligeramente a su organización de contenidos. Para ello extiende la especificación de IMS con algunos requerimientos y guías de implementación.

## Componentes del Empaquetamiento de Contenidos

Para conseguir la interoperabilidad de contenidos basados en Web con los sistemas y herramientas IMS define un conjunto de estructuras de datos que deberán seguir los paquetes de contenidos. Estas estructuras estándar definen un formato común para el intercambio y gestión de contenidos educativos.

La estructura propuesta consta de un fichero XML de manifiesto y el contenido, formado por todos los recursos que intervienen en la experiencia educativa. En el manifiesto se definen que estructura tiene el contenido educativo y la relación con los recursos electrónicos existentes en el paquete, es decir, con el contenido en sí. Además debe llevar el nombre manifest.xml y estar en la raíz del paquete, así como todos los ficheros de control (DTD, XSD).

El paquete representa “la unidad de aprendizaje”. Si tenemos un curso en un servidor EVE, este estará formado por uno o más paquetes, pudiéndose agregar o desagregar del curso por sí mismo. El paquete tendrá significado por sí mismo, aun fuera de los cursos, por lo que se puede utilizar como parte de un curso, o como un curso en sí. Por esta razón debe contener la información necesaria para ejecutarse por separado, sin depender de otros paquetes.

Los paquetes necesitan un formato en el que agrupar el manifiesto y los contenidos, lo que se conoce en SCORM como fichero de intercambio de paquetes (PIF), que será la representación física del Paquete de Contenidos. SCORM dicta que el PIF debe cumplir la especificación del formato de compresión de datos RFC 1951 que por ejemplo cumplen los archivos “.zip”.

## Componentes del Manifiesto

En el fichero de manifiesto encontramos la información relativa a los contenidos del paquete. Al estar localizada en un fichero permite a las herramientas y sistemas que utilicen el paquete obtener una descripción del mismo de una manera rápida.

El manifiesto contiene metadatos que describen el paquete, la organización del contenido, los recursos de los que dispone, y un conjunto de submanifiestos que describen las unidades de aprendizaje anidadas en caso de que existan.

### *Metadatos*

Contienen información relativa al paquete como contenido educativo independiente. Dan una descripción de las características del mismo con fines de búsqueda y reutilización. En la descripción se abordan particularidades tales como la Agregación de Contenidos, propósitos del paquete, público al que está destinado, etc. Esta información está ligada al paquete como unidad, y aunque puede dar descripciones de sus partes, está dirigida a dar información del paquete en concreto. Ya vimos que otros *Componentes del Modelo SCORM de Contenidos* llevan sus propios metadatos asociados.

### *Organizaciones*

Las Organizaciones sirven para describir la organización de la estructura en el paquete. Es una estructura jerárquica, ya que sigue la Especificación del Paquete de Contenidos de IMS, en la que no se sigue ninguna nomenclatura específica por lo que términos como “lección” o “tarea” quedan a disposición del desarrollador del contenido.

En la especificación de IMS se abordan los siguientes aspectos en cuanto a organizaciones de contenido se refiere:

- **Múltiples Organizaciones de Contenido:** la especificación de IMS permite que en un mismo paquete se definan distintas formas de organizar el contenido. Tiene sentido utilizar varias alternativas de organización para utilizar los mismos recursos en distintos contextos educativos, y se deberá seleccionar una como “por defecto” en caso de que coexistan varias para no sobrecargar los sistemas con la evaluación de cual de ellas utilizar. Por lo menos debe existir una descripción de organización. Que los recursos se encuentren separados de la forma de organizarlos permite la coincidencia de distintas organizaciones en un mismo paquete.
- **Organización del Contenido:** detalla una posible forma de uso de los recursos educativos presentes en el paquete. La Organización del Contenido permite al desarrollador dar un mapa de cómo deberá transcurrir la experiencia educativa. Explicará el orden y la forma de organizar el paquete en unidades de aprendizaje que engloban un conjunto de recursos. Puede organizarlo como una tabla de contenidos, en la que el alumno puede elegir entre ellos, o como un camino con distintas rutas en función de la actividad que desarrolle el alumno frente al material educativo. Esta estructura debe ser entendida por el EVE para que siga las indicaciones que dio el desarrollador, quedando en mano de cada sistema en concreto como lo representa internamente. Cabe destacar que solo se describe la estructura, y no los recursos en concreto que se utilizan. Estos quedan descritos en otra parte del manifiesto.
- **Representación de la Estructura del Contenido:** la organización de contenidos SCORM incluye los siguientes aspectos:
  - **Jerarquía de Contenidos:** es una representación del contenido de tipo árbol. Se deja abierta al creador de la experiencia educativa la elección de la metodología de diseño a utilizar, pudiendo representar los elementos de este árbol conceptos tales como “Curso” o “Capítulo”. IMS especifica que la raíz del

árbol se representa mediante un elemento Organización, y sus hijos quedan representados con un elemento Item, los cuales podrá anidar otros elementos Item. Cada Item puede referenciar a un elemento Recurso, y cada Recurso puede ser referenciado por más de un Item. Los Recursos representan un recurso electrónico que será ejecutado cuando se acceda al Item que lo referencia. Los perfiles de aplicación SCORM e IMS SS Specification restringen esta estructura en cuanto que solo los Item hoja podrán referenciar Recursos, y que la estructura debe tener por lo menos un Item. Además SCORM añade que los recursos deberán ser Assets o SCO.

- **Metadatos:** con estos metadatos se pretende proveer información de porqué un recurso se utiliza en esa parte de la estructura en concreto, mas que dar información acerca del recurso en sí. Por ello, se considera información dependiente del contexto. Esta información será de utilidad en caso de que se prevean cambios, reorganizaciones o disgregaciones del paquete.
- **Secuencia, Secuencia Adaptativa y Navegación:** hay varias posibilidades a la hora de presentar los contenidos al alumno. La más básica es que el alumno pueda elegir a que parte de la estructura de contenidos acceder o dejarse guiar por un recorrido lineal de la misma. Este es el comportamiento por defecto y se da cuando no hay información de Secuencia y Navegación. SCORM define un mecanismo de Secuencia y Navegación (en el correspondiente libro técnico, el cual se analizará mas adelante en este documento) mediante el cual se puede manejar el flujo de la experiencia educativa a lo largo de la estructura de contenidos del paquete. Con ello se pueden crear experiencias de aprendizaje adaptativas, presentando determinadas actividades en función del progreso del alumno, o de otras variables que se puedan almacenar en el EVE. La información de Navegación y Secuencia se asocia a los elementos del árbol de jerarquía los cuales representan actividades a las que el alumno se enfrentará, por lo que esta íntimamente ligada a la estructuración del paquete. Tal como dictan IMS SS Sequencing y SCORM, esta información aparece junto con la representación XML de las Organizaciones. Además, que aparezca definido en el manifiesto favorece que cualquier EVE compatible con SCORM interprete dicha información.

### *Recursos*

El componente “recursos” dentro del manifiesto representa contenidos electrónicos localizados tanto dentro del paquete como de una fuente externa que podrán ser utilizados en la ejecución de las actividades. Cada recurso individual podrá ser un Asset o un SCO, y podrá contener varios recursos electrónicos. Viene a ser un inventario de los recursos a los que puede referirse las Organizaciones.

### Contenido

Son los ficheros físicos contenidos en el paquete, o las referencias URI (Universal Resource Indicator) a ficheros externos. Es conveniente que todos los contenidos queden referenciados en el fichero de manifiesto por medio de su relación con el componente recursos visto con anterioridad para evitar problemas a la hora de intercambiar paquetes.

### **The SCORM Run-Time Environment RTE**

El libro técnico Entorno de Ejecución SCORM trata de abordar los aspectos que intervienen en la ejecución de contenido educativo (Assets y SCO) y la comunicación entre SCO y los entornos EVE. En él se encuentran las especificaciones y guías que deben seguir los EVE para ejecutar los contenidos y comunicarse con ellos a través de una API y un modelo de datos específicos.

A modo de resumen, a continuación detallamos los conceptos principales en los que se centra el contenido del libro técnico SCORM RTE. Cabe destacar que el proceso de selección del contenido educativo en función de información de navegación y secuencia queda fuera del alcance de este apartado, y se explica con detalle en el apartado correspondiente:

- Lanzamiento de contenido educativo, ya sea contenido que se puede comunicar o no con el EVE, definiendo una forma genérica de ejecutarlo en un entorno Web y un conjunto de responsabilidades que deben cumplir ambas partes cuando exista comunicación para una correcta gestión de la información intercambiada.
- Una API común para conocer el estado de la comunicación (conectado, desconectado, error, etc.) de forma que se pueda registrar y recuperar información en el EVE.
- Un modelo de datos para definir el formato de la información a compartir. El estado del modelo de datos debe ser persistente, es decir, mantenerse a lo largo de las sesiones de usuario mientras el gestor del EVE o del curso en el que se encuentra el contenido lo considere oportuno.

Al definir una forma normalizada en la conexión, ejecución y comunicación entre los sistemas de gestión de aprendizaje y los contenidos educativos se consigue amoldarse a la gran mayoría de los requerimientos SCORM comentados al principio de esta sección. Así un contenido que se comunique vía RTE con un EVE podrá ser utilizado en cualquier EVE que se amolde a esta especificación, consiguiendo así interoperabilidad y reutilización, se reducen los costes de creación de contenido al poder ser reutilizados (asequible), y se facilita su mantenimiento.

### **Gestión del Entorno de Ejecución**

El EVE se encarga de lanzar el contenido educativo que corresponda (la elección del contenido viene determinada por el modelo de navegación y secuencia) en un entorno Web y de almacenar y evaluar las acciones que realice el usuario sobre el contenido. Una vez se ha elegido el contenido, el EVE sabrá que recurso ejecutar a través de la estructura de contenido (jerárquica y en forma de árbol) tal y como vimos en el apartado anterior (CAM). Con un modelo de lanzamiento común se consigue estabilidad en el comportamiento del contenido educativo más allá de la implementación concreta del EVE.

### **Modelo Temporal del Entorno de Ejecución**

Cuando el EVE lanza un contenido educativo en la Web, debe realizar una serie de acciones para conseguir almacenar la actividad del alumno. El IEEE en su estándar para las tecnologías educativas 1484.11.1 “Modelo de Datos para la Comunicación con Objetos de

Contenido” define una relación temporal entre el usuario y el contenido que SCORM adopta:

- **Intento del Estudiante** (*Learner Attempt*): es el esfuerzo del estudiante por completar los objetivos propuestos en el contenido, y puede que necesite más de una sesión para llevarlo a cabo.
- **Sesión de Aprendizaje** (*Learner Session*): periodo en el que el estudiante está accediendo al contenido sin abandonarlo en una misma sesión.
- **Sesión de Comunicación** (*Communication Session*): periodo de conexión entre el contenido y la API.
- **Sesión de Login** (*Login Session*): tiempo que el usuario permanece en el sistema, desde que introduce sus datos de usuario hasta que abandona el sistema.

Cabe destacar que cuando se inicia un nuevo “Intento del Estudiante” el EVE debe crear e inicializar un nuevo conjunto de datos cuyo formato viene determinado por el modelo de datos que veremos más adelante. SCORM no impone ninguna restricción de lo que deba hacer el EVE con los intentos pasados, solo si se han suspendido, y será decisión de los creadores del EVE de almacenarlos para hacer históricos o descartarlos.

#### Lanzamiento de Objetos de Contenido

El proceso de lanzamiento de Assets y SCOs (contenidos definidos como “ejecutables” en el libro CAM) define una forma estándar de mostrar los contenidos en el navegador Web del estudiante. Es responsabilidad del EVE decidir la secuencia del contenido a mostrar en función de la información de la secuencia y navegación planteada en el paquete de contenidos como se define en el CAM y siguiendo las indicaciones del SN, en función de las cuales se elegirán la sucesión de actividades a mostrar teniendo en cuenta o no la interacción del usuario con el contenido.

También es responsabilidad del EVE elección de qué actividad se mostrará y ejecutar el contenido asociado a dicha actividad. Para ello utilizará la URI asociada a la actividad a ejecutar. Esta relación está definida en el fichero de manifiesto, tal y como vimos en el CAM. SCORM no indica ninguna manera específica de implementar el lanzamiento de contenidos, solo que este debe amoldarse al protocolo HTTP (Hypertext Transfer Protocol).

En cuanto a la naturaleza de los contenidos lanzados, SCORM solo requiere que los Assets sean lanzados vía HTTP. Para los SCOs encontramos más restricciones. Se requiere que el EVE coloque una instancia del API la cual debe ser accesible por DOM (Document Object Model) en una de las ventanas dependientes de la ventana principal que utiliza el EVE. El SCO está encargado de buscar la instancia de la API y de amoldarse a sus especificaciones. Una vez encontrada se podrá iniciar la comunicación entre ambas partes. Solo se puede lanzar un SCO por EVE y usuario.

#### **Interfaz de Programación de Aplicaciones (API)**

Tras los esfuerzos de la AICC en sus Guías para la Interoperabilidad (CMI001), el LTSC cogió el testigo continuando el trabajo y estandarizando muchos de los procesos abordados en el CMI. SCORM da una interpretación del uso de la API para los servicios de

comunicación en el entorno de ejecución (RTS) definido en el estándar<sup>9</sup> IEEE 1484.11.2-2003. El concepto de RTS en la perspectiva de SCORM está íntimamente ligado al EVE, ya que es quien provee el servicio. La API consiste en un conjunto de funciones que el EVE debe implementar para que el SCO las utilice con el objetivo de compartir y transmitir información mediante un conjunto de servicios ECMAScript, mas conocidos como JavaScript. Aunque SCORM no dicte como debe ser la implementación, si exige el cumplimiento de unas responsabilidades por parte del EVE y del SCO para conseguir un funcionamiento estándar. Con el uso de la API se cumplen los requerimientos de reutilización e interoperabilidad. Sólo el SCO utilizará las funciones de las API, tanto del intercambio de datos como del inicio y finalización de la comunicación.

Los métodos que proporciona la API se pueden agrupar en tipos:

- **Métodos de Sesión:** Se encargan del establecimiento y finalización de la comunicación entre el EVE y el SCO. Dentro de este grupo encontramos los siguientes métodos:
  - ***Initialize(“”)***: inicializa la comunicación. Devolverá “true” o “false” en función de que se haya podido o no establecer la comunicación.
  - ***Terminate(“”)***: finaliza la comunicación. La llamada a este método asegura la persistencia de los datos almacenados en el EVE. Devolverá “true” o “false” en función de que haya finalizado correctamente la comunicación, y por tanto la persistencia de datos.
  
- **Métodos de Transmisión de Datos:** Intercambian valores del modelo de datos entre el EVE y el SCO. Dentro de este grupo encontramos los siguientes métodos:
  - ***GetValue(“parámetro”)***: solicita una determinada información almacenada en el EVE. “parámetro” es la parte del modelo de datos a la que se desea acceder para recuperar la información. Nos devolverá una cadena de caracteres con el valor requerido. Si esta cadena llega vacía, habrá que comprobar si ha ocurrido un error comprobando el valor la función “*GetLastError()*” .
  - ***SetValue(“parámetro1”, “parámetro2”)***: envía información al EVE. Almacena la información contenida en “parámetro2” en la parte del modelo de datos del EVE especificada por “parámetro1”.
  - ***Commit(“”)***: asegura la persistencia de datos en el EVE. Si el EVE guarda en una cache la información intercambiada con el SCO, al realizar una llamada a la función “commit” se almacena en memoria física. Devolverá “true” si se realizó la operación con éxito y “false” en caso contrario.
  
- **Métodos de Soporte:** Se utilizan para otras tareas, como el soporte de errores. Dentro de este grupo encontramos los siguientes métodos:
  - ***GetLastError(“”)***: obtiene el código de error generado por la última llamada de la API, sin modificar el estado. Devuelve una cadena de caracteres con el código de error, en la que “0” refleja la ausencia de error.
  - ***GetErrorString(“”)***: obtiene una descripción textual del error generado por la última llamada de la API, sin modificar el estado. El contenido del texto es

---

<sup>9</sup> IEEE 1484.11.2-2003 Standard for Learning Technology - ECMAScript Application Programming Interface for Content to Runtime Services Communication

responsabilidad del EVE, y devolverá una cadena de caracteres vacía en caso de no haber definido ninguna descripción para un error.

- **GetDiagnostic("parameter")**: está específicamente diseñada para que los creadores del EVE definan información adicional. "parameter" será el identificador que hayan asociado a una información en concreto, y su relación debe ser explicada por los desarrolladores del EVE.

En el RTE se encuentra la descripción detallada de los códigos de error existentes, y da la posibilidad de añadir la definición de códigos nuevos al EVE. Cabe destacar que SCORM v1.2 tiene los mismos métodos pero con otra nomenclatura: se le añade el prefijo "LMS" (LMSGetValue, LMSSetValue), y el método "Terminate" se llama "LMSFinish".

IEEE define en el estándar 1484.11.2-2003 un modelo conceptual de los posibles estados y transiciones por los que puede pasar la instancia de la API a lo largo de una ejecución. El modelo no tiene por que ser implementado, sino que surge con el ánimo de explicar en qué determinada situación de la comunicación entre EVE y SCO se pueden invocar a cada uno de los métodos que brinda la API.

- **No inicializado** (*Not Initialized*): antes de que se realice con éxito la llamada de establecimiento de la comunicación "Initialize()".
- **En funcionamiento** (*Running*): mientras se mantenga la comunicación entre el SCO y el EVE. Puede definirse como el estado entre una llamada "Initialize()" y una "Terminate(())".
- **Finalizado** (*Terminated*): una vez termina la comunicación tras una llamada a "Terminate()".

### Responsabilidades del EVE y del SCO ante el RTE

El modelo de aplicación SCORM da una serie de requerimientos a los desarrolladores de EVE y SCO a la hora de proveer y utilizar la instancia de la API. Analicemos las responsabilidades de ambas entidades.

El EVE debe lanzar la instancia de la API y el SCO. La localización de la API debe ser accesible vía DOM desde la localización del SCO. El SCO es el responsable de encontrar la instancia de la API vía JavaScript. En cuanto a la accesibilidad de la API, esta debe estar en la cadena de padres de la ventana en donde se encuentra el SCO. Si se lanza en una ventana a parte, el SCO deberá buscar en la cadena de padres de la ventana. Aunque SCORM no fija la forma de implementar la búsqueda de la instancia de la API, se debe realizar en las localizaciones expuestas. Además propone un código JavaScript para ello, el cual está definido en el estándar IEEE. Una vez establecida la comunicación entre ambos, el SCO tiene la obligación de invocar como mínimo a "Initialize()" y "Terminate()".

### **Modelo de datos del entorno de ejecución**

En el modelo de aplicación SCORM se decide definir un modelo de datos común para de manera que se pueda almacenar determinada información de los SCO en EVE independientemente de su tipo. Esta información incluye datos sobre la interacción del usuario con el contenido y datos sobre el usuario y no es cerrada, por lo que los autores de contenidos educativos pueden definir nuevas partes del mismo. El modelo de datos SCORM del entorno de ejecución (The SCORM Run-Time Environment Data Model) de

la versión 2004 está basado en el estándar para el modelo de datos de comunicación de contenidos educativos<sup>10</sup> definido por IEEE LTSC CMI. En versiones anteriores de SCORM el modelo de datos estaba basado en el estándar de AICC CMI001 “Guidance for Interoperability”, basándose hoy en día el que presenta IEEE.

Cada campo de los registros del modelo de datos tiene un tipo definido. Los tipos de datos que se soportan en SCORM 2004 son cadenas de caracteres (characterstring), cadenas de caracteres localizadas, es decir, que indican el lenguaje utilizado (localized\_string\_type), tipo de lenguaje, que sirve para identificar un lenguaje (languaje\_type), identificadores cortos y largos, que usan la sintaxis definida por la Universal Resource Identifier (URI) (long\_identifier\_type y short\_identifier\_type), los números enteros (integer), el tipo estado que define un conjunto de posibles valores o vocabulario (state), los números reales con precisión de siete dígitos (real (10,7)), el tipo tiempo que representa fechas (time (second,10,0)) y los intervalos de tiempo (timeinterval(second,10,2)).

A continuación se muestran todos los registros que posee el modelo de datos de SCORM 2004:

**Tabla 1: Modelo de datos CMI para SCORM 2004**

Elemento del Modelo de Datos	Notación	Descripción
Comments From Learner (Comentarios del Alumno)	cmi.comments_from_learner	Almacena información proveniente del usuario sobre la experiencia educativa.
Comments From LMS (Comentarios del LMS)	cmi.comments_from_lms	Información para ser mostrada al alumno. Permite a los creadores de contenidos añadir información adicional para los alumnos a los que está pensado el contenido educativo.
Completion Status (Nivel del Completitud)	cmi.completion_status	Nivel de completitud del SCO. Sus valores están comprendidos dentro de un vocabulario (“completed”, “incomplete”, “not attempted”, “unknown”). Tomará cada uno de los valores en función de otros campos del modelo de datos (“Completion Threshold”, “Progress Measure”).
Completion Threshold (Umbral de Completitud)	cmi.completion_threshold	Define un umbral a partir del cual se considera completado el SCO
Credit (Credito)	cmi.credit	Controla si el rendimiento del alumno (las interacciones almacenadas en este modelo de datos) será informativo o podrá ser utilizado para el secuenciamiento. Está estrechamente ligado con el campo “Mode”. Puede tener los valores “credit” o “no-credit”.
Entry (Acceso)	cmi.entry	Información relativa a si el alumno ha accedido previamente al contenido educativo en el “Intento del estudiante” actual. Puede tomar los valores “ab-initio”, “resume” o “” en función de si accedió o no previamente.
Exit (Salida)	cmi.exit	Almacena el motivo del abandono del SCO por parte del alumno. Puede tomar los valores “suspend”

<sup>10</sup> 1481.11.1 Standard for Learning Technology – Data Model for Content Object Communication

		(indica que el “Intento del alumno” no ha terminado), “normal”, “logout”, “time-out” o “” (indicando la causa del abandono del SCO en estos casos).
Interactions (Interacciones)	cmi.interactions	Almacena información relativa a las interacciones del usuario con el SCO.
Launch Data (Datos de Inicialización)	cmi.launch_data	Datos específicos para la inicialización del SCO en un contexto específico.
Learner Id (Identificador del Alumno)	cmi.learner_id	Identificador del alumno que ejecuta el SCO.
Learner Name (Nombre del Alumno)	cmi.learner_name	Nombre del alumno, provisto por el LMS.
Learner Preference (Preferencias del Alumno)	cmi.learner_preference	Preferencias del alumno en relación del uso del SCO, como el lenguaje o el nivel del audio.
Location (Localización)	cmi.location	Refleja la ubicación en el SCO. Su uso esta pensado para marcar un punto de reentrada o un punto de chequeo de alguna característica.
Maximum Time Allowed (Máximo Tiempo Permitido)	cmi.max_time_allowed	Refleja el tiempo máximo que el aprendiz puede usar el SCO.
Mode (Modo)	cmi.mode	Indica si las acciones del alumno en el SCO tendrán valor meramente informativo o serán validas para analizarlas con distintos fines, como la evaluación. Tiene un vocabulario definido el cual comprende los términos observación (“browse”) y revisión (“review”), los cuales son informativos, y normal (“normal”), para el cual si tendrá sentido analizar lo que se almacene en el modelo de datos. Tiene relación con el campo “Credit”, para los dos primeros valores de “Mode” tomará el valor “no-credit” y para el tercero tomará el valor “credit”.
Objectives (Objetivos)	cmi.objectives	Almacena objetivos relacionados con el SCO.
Progress Measure (Medida del Progreso)	cmi.progress_measure	Almacena una medida numérica del progreso del alumno para completar el SCO. Está relacionado con el campo “Completion Status”.
Scaled Passing Score (Escala de Puntuación)	cmi.scaled_passing_score	Identifica la escala de puntuación para un SCO para que quede en el rango (-1,1).
Score (Puntuación)	cmi.score	Almacena la puntuación global del alumno.
Session Time (Tiempo de Sesión)	cmi.session_time	Tiempo empleado en la sesión actual.
Success Status (Estado de Éxito)	cmi.success_status	Indica si el alumno ha llegado a dominar el SCO. Tiene un vocabulario definido que comprende los términos aprobado (“passed”), suspenso (“failed”) y desconocido (“unknown”).
Suspend Data (Datos de Suspensión)	cmi.suspend_data	Almacena información creada por el SCO como resultado de los accesos o interacción del alumno cuando se suspende una sesión. Su uso es el de intercambiar información entre sesiones de usuario.
Time Limit Action	cmi.time_limit_action	Da indicaciones al SCO cuando se

(Tiempo Límite de Acción)		alcanza el tiempo límite. Tiene un vocabulario definido en el que se refleja si se podrá continuar o no una vez se alcance el límite de tiempo y si se mostrará o no un mensaje al alumno. Sus posibles valores son “exit,message”, “exit,no message”, “continue,message”, “continue,no message”.
Total Time (Tiempo Total)	cmi.total_time	Almacena el tiempo total, sumando el tiempo de todas las sesiones de usuario.

Analicemos con más detalle algunos de los registros que son complejos (más de un campo) y que consideramos más importantes dentro del modelo de datos.

### Interacciones

En este registro se almacenan las interacciones que el usuario realiza con el contenido educativo. Entendemos por interacciones todas aquellas acciones que realice el alumno en el contenido educativo que el autor del SCO pretenda que queden registradas. Los usos más típicos pueden ser respuestas a determinadas preguntas o la consecución de alguna tarea que tenga un valor especial. Este registro queda implementado como un array en el que se almacenan todas las interacciones que el SCO mande almacenar. De esta manera, cada entrada del array queda especificada por `cmi.interactions.n`, y se puede obtener el recuento de los elementos existentes invocando a la función de la API “GetValue” del campo `cmi.interactions._count` (`GetValue(“cmi.interactions._count”)`). Para recuperar o dar valor a cualquiera de los campos del registro que veremos a continuación, simplemente debemos realizar las siguientes llamadas de la API (para el campo “id” y la interacción situada en la posición 4, por ejemplo): “GetValue(“cmi.interactions.3.id”)” o “SetValue(“cmi.interactions.3.id”,”objetive4”)”. Cabe destacar que para acceder a la posición 4 ponemos el índice 3 ya que el array empieza en 0.

Hay dos posibles formas de que el SCO interprete las interacciones: de publicación (journaling) o de estado (status). En la primera cada interacción será almacenada como un nuevo elemento del array, de manera que habrá un registro dentro del array por cada interacción. En la segunda, las interacciones se actualizan, es decir, si se realiza una interacción que ya se había realizado no se añade un nuevo registro, sino que pasa a ser el registro de la última vez que se realizó dicha interacción. Cabe destacar que será el autor del SCO quien debe conocer estas dos posibles interpretaciones y decidir cual va a usar.

Dentro de los campos que tiene cada uno de los registros interacción que se van a almacenar en el array nos encontramos con el identificador (`cmi.interactions.n.id`) que será la etiqueta que diferencie una interacción en concreto. Este campo será el que nos sirva para actualizar el registro en función de la interpretación dada, como explicamos con anterioridad. Otro campo importante es el de tipo (`cmi.interactions.n.type`) y sirve para conocer como deberá ser interpretada la respuesta asociada a la interacción. Este campo tiene un vocabulario definido. Así, algunos de los tipos que encontramos son tipo verdadero-falso (“true-false”), selección de un conjunto de opciones de una escala (“likert”), identificación de una secuencia lógica de elementos (“sequencing”) o de respuesta corta (“fill-in”). El resto de tipos son “choice”, “long-fill-in”, “matching”, “performance”, “numeric” y “other”. En relación con el tipo encontramos el campo de respuestas correctas (`cmi.interactions.n.correct_responses`). Es también un array en el que se almacenan el conjunto de patrones que validarán la respuesta dada en la interacción

(cmi.interactions.n.correct\_responses.n.pattern). Depende del tipo, así para las de tipo verdadero-falso será simplemente almacenar si la respuesta correcta es “true” o “false”, o para las de tipo “matching” se almacenará la fuente y el destino en los conceptos a unir. Consecuentemente, es importante reseñar también el campo cmi.interactions.n.lerner\_responses (respuestas del alumno), la cual también depende del tipo que se le haya asignado a la interacción y para el cual SCORM define un patrón de cómo debe ser la respuesta.

Otro campo interesante es el de objetivos (cmi.interactions.n.objectives). En este campo quedan reflejados los identificadores de objetivos que se han cumplido para la interacción almacenada, por lo que queda implementado como un array (cmi.interactions.n.objectives.m.id). Este identificador está relacionado con el que encontraremos en el registro “Objetivos” que también se analiza en esta sección, pero no es condición indispensable que este dentro de dicho registro.

## Objetivos

En este registro del modelo de datos se permite a los autores de los contenidos educativos almacenar información al respecto de distintos objetivos que pueden aparecer a lo largo del desarrollo del contenido. De igual manera que las Interacciones, los Comentarios del LMS y los Comentarios del Alumno, se implementa como un array en el que se almacenan todos los objetivos asociados a un SCO, quedando especificada cada entrada del array por cmi.objectives.n, pudiendo recuperar el número de elementos o un campo de un objetivo en concreto de igual manera que la vista para Interacciones.

El primer campo de este registro que tiene importancia es el identificador (cmi.objectives.n.id), ya que servirá para diferenciar los distintos objetivos y, en algunas ocasiones, relacionará un objetivo del modelo de datos con un objetivo de navegación y secuencia. Además, los objetivos tienen los campos Puntuación, Estado de Éxito, Nivel de Completitud, y Medida del Progreso como los vistos en el modelo de datos. Tienen el mismo significado, pero en vez de estar asociados al contenido educativo en sí (SCO), dan medidas del avance del alumno en relación al objetivo en concreto.

Este registro tiene una importancia especial puesto que se puede usar con propósitos de navegación y secuencia. Entraremos más en detalle en estos términos en el análisis del libro SCORM que trata de este tema, pero introducimos navegación y secuencia como el mecanismo que propone SCORM para el flujo de las subactividades contenidas en un contenido SCORM y estructuradas como explica SCORM CAM. Un SCO referencia objetivos a través de los identificadores de igual manera que la información de secuencia se asocia a una actividad de aprendizaje. Por ello SCORM define como utilizar los identificadores para relacionar los objetivos definidos con propósitos de secuencia con los del modelo de datos. Para ello, SCORM obliga a un EVE que sea SCORM compatible que cree en el modelo de datos un objetivo por cada elemento hijo de <imsss:objectives> en el manifiesto, inicializando el identificador con el valor del atributo ID de dicha etiqueta, y los valores Nivel de Completitud y el campo de Puntuación “escala” con la información “” asociada al objetivo.

Las principales diferencias entre el modelo de datos SCORM v1.2 y SCORM 2004 residen en que en la primera se agrupan bajo la nomenclatura “cmi.core” los campos “student\_id”, “student\_name”, “lesson\_location”, “credit”, “lesson\_status”, “completion\_status”, “entry”, “score”(y los campos que lo forman), “total\_time”, “lesson\_mode”, “exit”,

“session\_time”. Además se elimina el campo “student\_data”. Existen diferencias en los tipos y vocabularios soportados en ambas versiones.

### **The SCORM Sequencing and Navigation SN**

Los contenidos compatibles con SCORM pueden organizarse en actividades. El libro **Secuencia y Navegación SCORM** facilita a los desarrolladores de contenidos un mecanismo que permite describir el orden o flujo de las actividades, es decir, como serán mostradas al alumno. Este mecanismo se conoce con el término de **Secuenciamiento**. Además SCORM suministra un mecanismo para permitir al alumno moverse por las actividades que conforman el contenido, conocido como **Navegación**. El Secuenciamiento indica que opciones de navegación estarán disponibles mientras se ejecuta una actividad en concreto. La Navegación depende del Secuenciamiento en cuanto que puede condicionar sus opciones de libre movimiento por el contenido. La diferencia fundamental entre Navegación y Secuenciamiento recae en que la Navegación permite al usuario elegir la actividad a través de una interfaz implementada en el EVE y en que el Secuenciamiento es un mecanismo pensado para que el desarrollador del contenido decida como se ejecutará la secuencia de actividades, pudiendo depender o no de la interacción del usuario.

El libro SCORM SN tiene una estrecha relación al libro SCORM CAM. Ya vimos en la descripción del CAM que los contenidos SCORM pueden incluir una descripción de su estructura en el fichero de manifiesto. Aquí se puede encontrar información para que el EVE interprete y gestione la secuencia de actividades.

La comunicación entre el contenido (SCO) y el EVE permite almacenar el rendimiento del alumno y condiciona el flujo de la secuencia de actividades. Por tanto, las acciones de secuencia de contenidos están ligadas a la interacción y el progreso del estudiante en las actividades. En función de las acciones concretas que realice en una actividad se permitirá o no elegir la siguiente actividad (que pueden ser también actividades anteriores) o se obligará a ejecutar una acción en concreto. De esta manera se permite realizar una adaptación de los contenidos en función de la actividad realizada por el alumno. Esta relación es una de las claves de la secuencia de contenidos, por lo que se analiza con detalle en este libro técnico.

El Secuenciamiento explicado en este libro técnico está basado en la especificación IMS SS Simple Sequencing (SS). SCORM da una interpretación de la misma que aborda como utilizarla. Recordemos que SCORM es un perfil de aplicación.

### **Conceptos necesarios para entender el Secuenciamiento**

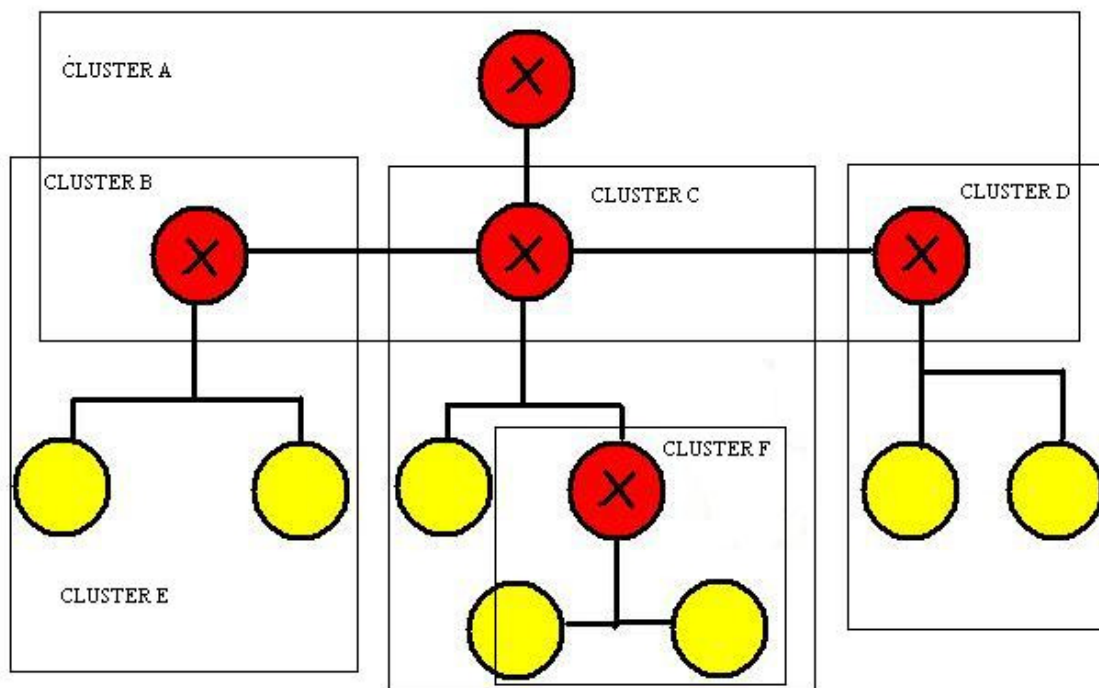
En el archivo de manifiesto se puede definir una organización jerárquica que nos da una idea de cómo se compone el contenido en distintas actividades y subactividades. De esta manera se permite al desarrollador representar cursos, lecciones, unidades, etc. La estructura que representa la jerarquía de contenidos en Secuenciamiento esta muy ligada con la estructura explicada en CAM.

En el Secuenciamiento el término **Actividad** se utiliza para todo elemento representado como Ítem en el fichero de manifiesto. Vemos que tiene el mismo significado que el concepto Actividad en CAM. En esta sección nos referiremos a Ítem cuando hablemos del concepto de Actividad en CAM. La estructura jerárquica de las Actividades viene dada por

el **Árbol de Actividades**. El Secuenciamiento utiliza dicha estructura para tener una representación de la relación padre-hijo existente entre Actividades.

Dentro de los Árboles de Actividades se pueden localizar subestructuras que engloban una actividad padre y sus hijos inmediatos. Estas subestructuras se denominan **Clusters** y representa el mismo concepto denominado Agregación en CAM. Tienen importancia ya que elementos del Modelo de Datos de Secuencia se aplican a estas subestructuras. Además cada elemento padre en un Cluster lleva asociada la información de secuencia para evaluar la ramificación entre sus hijos, y en caso de que estos sean hojas, ejecutar el contenido que llevan asociado. El Árbol de Actividades se corresponde con la Organización de Contenidos de un manifiesto (explicada en CAM) y se construye a partir de él..

El Árbol de Actividades puede tener determinada información de secuencia adicional para definir y condicionar el flujo del contenido. Dicha información está asociada a cada Actividad y se define en el fichero de manifiesto. En caso de no existir, se utilizará la jerarquía del contenido para mostrar los contenidos de una manera plana, es decir, uno detrás de otro. Además de la información de secuencia, las Actividades contienen **datos de seguimiento** para representar su estado en cada momento.



**Figura 2: Representación del Árbol de Actividades y su descomposición en clusters.**

Otro concepto que tiene relación con el Árbol de actividades es el de **Intento**. Un Intento comienza cuando se ejecuta un contenido (Asset o SCO) en una actividad hoja y dura hasta que el alumno no lo termine, ya sea por finalización de la tarea o por abandono. Mientras un Intento transcurre se pueden satisfacer objetivos del modelo de datos (vía RTE). El estado de seguimiento de la Actividad solo puede cambiar una vez ha terminado el Intento. El concepto aquí analizado tiene relación con el Intento del Estudiante analizado en el libro RTE.

Siguiendo con las comparaciones con el CAM, una **Actividad de Aprendizaje** en SN se corresponde con el concepto de Agregación de Contenido. Es una unidad conceptual con

un determinado objetivo, que a su vez puede estar formada por otras Actividades de Aprendizaje, y en cuyas hojas se ejecutan los Recursos (Assests o SCO).

Las Actividades pueden tener **Objetivos** asociados. Estos objetivos en términos educativos son objetivos de aprendizaje, o determinadas metas dentro de la experiencia educativa que el autor de los contenidos desea que sean asimilados y superados por el alumno. El concepto de Objetivo en SN es el mismo que en el modelo CMI, pero son entidades separadas, es decir cada mecanismo utiliza las suyas propias, aunque tiene sentido que estén relacionadas. Una gran diferencia entre estos dos conceptos es que un SCO tiene su propio estado de satisfacción y una valoración y cada objetivo del SCO tiene sus propias medidas de satisfacción. Sin embargo una Actividad solamente almacena la medida de completitud, mientras que la medida de satisfacción es propia de cada Objetivo. Aparece entonces el concepto de Objetivo primario en el que se almacenan una medida de la satisfacción para la Actividad.

Los Objetivos son locales a cada Actividad. Pero los Objetivos también pueden ser compartidos por varias actividades, es decir, ser **Objetivos Globales**. Un Objetivo local solo puede obtener información de un Objetivo Global, y entre todos los Objetivos asociados a una actividad, solo uno de ellos puede escribir en el mismo Objetivo Global.

El Secuenciamiento utiliza la información recogida de los tres siguientes modelos para decidir el flujo de actividades mostradas al usuario:

- **Modelo de Seguimiento** (“*Traking Model*”): almacena información de la interacción del usuario con la actividad.
- **Modelo de Estado de la Actividad** (“*Activity State Model*”): es un subconjunto del modelo de seguimiento formado por la información del estado de la actividad y la información del estado global. Gestiona el estado de la secuencia para cada actividad y el estado global del Árbol de Actividades en tiempo de ejecución y es usado por la implementación del Secuenciamiento del EVE para llevar a cabo la gestión del Árbol.
- **Modelo de Definición de Secuencia** (“*Sequencing Definition Model*”): es utilizado por los procesos de secuencia para interpretar el modelo de seguimiento y conseguir la secuencia de actividades descrita por los comportamientos de secuencia definidos por el creador de los contenidos en el Paquete de Contenidos SCORM.

### **Modelo de seguimiento**

El **modelo de seguimiento** refleja el estado de las Actividades en cada momento. Aunque este conjunto de datos es propio del Secuenciamiento, está muy relacionado con el modelo de datos del RTE. De hecho, el modelo de seguimiento es un subconjunto del citado modelo con una terminología y tipos de datos distintos. Los campos del modelo del RTE que afectan al estado de una Actividad son “*completion\_status*”, “*satisfaction\_status*”, “*progress\_measure*” y “*score*”. Por tanto, las modificaciones que realiza el SCO en estos campos afectan a los datos de seguimiento. De esta manera se podrá condicionar la secuencia de contenidos en función de la interacción del alumno con el SCO.

Cada Actividad tiene un modelo de datos de seguimiento. Sin embargo solo las Actividades que ejecuten un SCO tendrán asociado un modelo de datos del RTE. Las Actividades que tienen un Asset no pueden recoger datos del RTE. Como los Assets no tienen la capacidad

de comunicarse con el EVE, este rellenará dicha información basándose en comportamientos y en valores por defecto definidos por el desarrollador. Por último, las Actividades que se corresponden con un Cluster rellenarán su modelo de seguimiento a partir del modelo de sus actividades hijas en un proceso conocido como **Rollup**.

El modelo de seguimiento tiene cuatro conjuntos que almacenan la información del estado del seguimiento:

- **Información del Progreso del Objetivo** (“*Objective Progress Information*”): muestra el progreso del alumno para cada uno de los objetivos de aprendizaje asociados a la actividad. Sus elementos se pueden agrupar por parejas, indicando uno de ellos el dato almacenado y otro si ese dato es válido. De esta manera, encontramos los elementos “Estado de Satisfacción del Objetivo” que indica si se ha conseguido el propósito del objetivo y “Estado del Progreso del Objetivo” que indica si el anterior es válido; y “Medida Normalizada del Objetivo” que indica con valor normalizado entre [-1.0..1.0] una medida del progreso del objetivo y “Estado de la Medida del Objetivo” que indica si el anterior es válido.
- **Información del Progreso de la Actividad** (“*Activity Progress Information*”): es la información que da una medida del estado del avance del alumno en la Actividad. Almacena información relativa a todos los Intentos en dicha actividad. Tiene un elemento que valida toda la información almacenada, conocido como “Estado del Progreso de la Actividad”. La “Duración Absoluta de la Actividad” almacena el tiempo total (todos los Intentos) que ha empleado el alumno en la Actividad, medido en segundos. La “Duración de la Experimentada en la Actividad” refleja el tiempo acumulado desde el inicio hasta el final, sin contar periodos de suspensión. La “Cuenta de los Intentos de la Actividad” almacena el número total de Intentos para la Actividad.
- **Información del Progreso del Intento** (“*Attempt Progress Information*”): almacena información relativa a cada Intento que realiza el alumno en una Actividad. Tiene un elemento que valida toda la información almacenada, conocido como “Estado del Progreso del Intento”. La “Cantidad de Finalización del Intento” es una medida del progreso del alumno en el intento entre [0.0..1.0], siendo 1 el indicador de que se ha finalizado el intento (esta relacionado con `cmi.progress_status`), y “Estado de Finalización del Intento” es un booleano que indica si finalizó o no. La “Duración Absoluta del Intento” y la “Duración de Experimentada en el Intento” reflejan lo mismo que en la “Información del Progreso de la Actividad”, pero para el Intento.
- **Información del Estado de la Actividad** (“*Activity State Information*”): almacena información del estado de todas las Actividades del Árbol para permitir los comportamientos de secuencia. El elemento “La Actividad está Activa” indica que se está realizando un Intento en la Actividad o en alguno de sus antepasados. El elemento “La Actividad está Suspendida” indica la suspensión de la Actividad y que almacene los datos de seguimiento del último Intento. Ambos valores son booleanos. El elemento “Hijos Disponibles” es una lista que indica el orden de los hijos disponibles de la actividad.

Además, se pueden guardar datos extra sobre el estado del Árbol de Actividades, identificando esta información como “**Información del Estado Global**”. Tiene dos elementos que almacenan actividad que se está ejecutando en el momento actual (“Actividad Actual”) y la actividad en la que se realizó la suspensión mediante una petición “Suspendir Todo” (“Actividad Suspendida”).

## El Modelo de Definición de Secuenciamiento

El **modelo de definición de Secuenciamiento** SCORM toma como base el de IMS SS al que añade algunos elementos específicos en términos de comportamiento y restricciones. Este modelo define un conjunto de reglas de Secuenciamiento a través del fichero de manifiesto que sirven para que los desarrolladores puedan añadir de comportamientos de secuencia en los contenidos. A continuación analizamos las distintas categorías en las que se pueden agrupar estas reglas. Las reglas pueden ser aplicadas a Clusters y/o a hojas dentro del Árbol de Actividades, y tienen un valor por defecto. En este documento vamos a indicarlo así: {cluster|hoja|ambos ; valor inicial}.

### Modos del Control de Secuencia

Controlan las opciones de Navegación. Permiten a los desarrolladores definir que actividades podrán ser ejecutadas. Existen los siguientes:

- **Sequencing Control Choice** {clusters;cierto}: el alumno puede elegir cualquier actividad del Cluster sin restricción ninguna.
- **Sequencing Control Choice Exit** {ambos;cierto}: indica si una actividad puede terminar cuando con petición de Navegación de Elección, es decir, cuando el alumno selecciona una actividad en la tabla de contenidos que se muestra en el navegador web.
- **Sequencing Control Flor** {clusters;falso}: cuando está activado permite que aparezcan los botones de Navegación “Siguiente/Previo” para los elementos hijos del Cluster.
- **Sequencing Control Forward Only** {clusters;falso}: restringe el acceso a las Actividades pasadas. Por tanto deberán quedar deshabilitadas las opciones de Navegación “Previo” y las peticiones de “Elección” que tengan como objetivo una actividad previa.
- **Use Current Attempt Objective Information** {clusters;cierto}: en caso de ser falso, se tendrá en cuenta la información de satisfacción recogida en la actividad actual durante Intentos pasados para contribuir al estado de Rollup de su actividad padre.
- **Use Current Attempt Objective Information** {clusters;cierto}: en caso de ser falso, se tendrá en cuenta la información de completitud recogida en la actividad actual durante Intentos pasados para contribuir al estado de Rollup de su actividad padre.

### Controles de Obligación de Elección

Limitan las Actividades que pueden estar presentes en la tabla de contenidos que permite al usuario elegir las actividades.

- **Constraint Choice** {cluster;falso}: para su valor inicial los controles de Navegación están limitados únicamente al botón “Siguiente”.
- **Prevent Activation** {cluster;falso}: no permite elegir una Actividad que no sea hija de la Actividad que se encuentra en ejecución.

## Descripción de Reglas de Secuencia

Permite definir reglas que controlen la secuencia teniendo en cuenta la información de seguimiento almacenada para la Actividad. La estructura de las reglas es la siguiente: if [conjunto\_condiciones] then [acción]. Cada regla esta formada por un conjunto de condiciones (una o más). Todas las condiciones y acciones que vamos a mostrar se pueden aplicar para Clusters y hojas, menos la acción “Salida” que sólo se puede aplicar a hojas.

Pasamos a analizar las **condiciones** que se pueden encontrar en las reglas:

- Reglas que comprueban valores del Objetivo asociado a la actividad en donde está definida la regla:
  - **Satisfacción** (*Satisfied*): comprueba si son ciertos el “Estado de Progreso del Objetivo” y el “Estado de Satisfacción del Objetivo”.
  - **Conocimiento del Estado del Objetivo** (*Objective Status Known*): cierto si el “Estado de Progreso del Objetivo” es cierto.
  - **Conocimiento de la Medida del Objetivo** (*Objective Measure Known*): cierto si el “Estado de la Medida del Objetivo” es cierto.
  - **Medida del Objetivo Mas Grande que** (*Objective Measure Greater Than*): comprueba que sea cierto el “Estado de la Medida del Objetivo” y que la “Medida Normalizada del Objetivo” sea mayor a la “Medida Umbral de la Condición de una Regla”.
  - **Medida del Objetivo Mas Pequeña que** (*Objective Measure Less Than*): comprueba que sea cierto el “Estado de la Medida del Objetivo” y que la “Medida Normalizada del Objetivo” sea menor a la “Medida Umbral de la Condición de una Regla”.
  
- Reglas que comprueban valores del Intento en el que se está desarrollando la actividad:
  - **Completado** (*Completed*): comprueba que el “Estado del Progreso del Intento” y que el “Estado de Finalización del Intento” sean ciertos.
  - **Conocimiento del Progreso de la Actividad** (*Activity Progress Known*): comprueba que el “Estado del Progreso de la Actividad” y que el “Estado del Progreso del Intento” sean ciertos.
  - **Intentado** (*Attempted*): comprueba si el “Estado del Progreso de la Actividad” es cierto y que el “Contador de Intentos de la Actividad” es positivo.
  - **Excedido el Límite de Intentos** (*Attempt Limit Exceeded*): comprueba que sean ciertos el “Estado del Progreso de la Actividad” y el “Condición de Límite – Control de Intento” y que el “Contador de Intentos de la Actividad” sea mayor o igual al “Condición de Límite – Límite de Intentos”.
  
- Otras:
  - **Siempre** (*Always*): se evalúa siempre como cierto.

Existe un conjunto de operadores para modificar las condiciones. La **Medida Umbral** de la Condición de la Regla es un valor real con cuatro dígitos decimales comprendido entre -1

y 1 definida por el creador del contenido y cuyo valor por defecto es 0. También encontramos **Combinación de Condiciones**, que permite combinar condiciones de dos formas, que todos los valores deban ser ciertos para que la condición se considere válida (All, actúa como el operador lógico AND) o que con que se cumpla una condición ya se considera como válido para ejecutar la acción asociada (Any, actúa como el operador lógico OR). Existen dos operadores lógicos unáricos (**Operadores de Condición**) que se pueden usar en conjunción con las condiciones cambiando su significado. Estos son el operador “Not” que cambia el valor resultante de la evaluación y “NO-OP”, que no realiza ningún cambio a la evaluación de la condición, y es el considerado por defecto en caso de que no haya ningún operador lógico.

Existen tres tipos de **acciones** asociadas a las reglas. Estas acciones se llevarán a cabo determinados comportamientos de secuencia en caso de que el resultado de evaluar las condiciones sea cierto. Vemos los tipos:

- **Acciones Precondición:** Se aplican mientras se recorre el Árbol de Actividades para elegir una actividad para ejecutar. Dentro de este tipo de acciones encontramos:
  - **Saltar** (*Skip*): Salta la Actividad a menos que sea directamente seleccionada y no permite seleccionarla mediante una petición de secuencia del modo “Flujo”.
  - **Desactivado** (*Disabled*): No se permite ejecutar la Actividad, y no permite las peticiones de secuencia “Flujo” y “Elección”.
  - **Escondida para Elección** (*Hidden from Choice*): la actividad no puede ser objetivo de una petición de secuencia del modo “Elección”, escondiéndola de la tabla de contenidos mostrados para Navegación.
  - **Parar Recorrido Hacia delante** (*Stop Forward Traversal*): impide a las Actividades posteriores en recorrido de preorden del Árbol ser elegidas a ejecutar.
- **Acciones Postcondición:** Se pueden aplicar cuando termina el Intento de una Actividad. Encontramos las siguientes acciones de este tipo:
  - **Salir del Padre** (*Exit Parent*): Procesa una petición de terminación de este tipo y la propaga a su padre.
  - **Salir de Todas** (*Exit All*): Procesa una petición de terminación de este tipo, la propaga hasta la raíz del árbol y devuelve una petición de secuencia “Salida”.
  - **Volver a Intentar** (*Retry*): devuelve una petición de secuencia de este tipo.
  - **Volver a Intentar Todas** (*Retry All*): Procesa una petición de secuencia “Salir de Todas” y devuelve una petición de secuencia “Comenzar”. Se vuelve a ejecutar el Árbol de Actividades desde la raíz.
  - **Continuar** (*Continue*): devuelve una petición de secuencia de este tipo.
  - **Previa** (*Previous*): devuelve una petición de secuencia de este tipo.
- **Acciones de Salida:** Se pueden aplicar después de que termine el Intento de una actividad hija. Encontramos las siguientes acciones de este tipo:
  - **Salida** (*Exit*): termina la actividad incondicionalmente.

## Condiciones de límite

Permite al creador del contenido limitar el número de veces que se puede ejecutar una Actividad. Pueden ser aplicados a Clusters y hojas Hay dos tipos:

- **Límite de Intentos:** permite limitar el número de intentos para una actividad. Tiene dos elementos: Condición de Límite – Control de Intento (“*Limit Condition Attempt Limit Control*”) que indica si el número de intentos para una actividad ha sido establecido; y Condición de Límite – Límite de Intentos (“*Limit Condition Attempt Limit*”) que indica el número de intentos disponibles para una actividad.
- **Duración Absoluta del Intento:** permite limitar el tiempo que puede emplear el alumno en un Intento de una actividad. Tiene dos elementos: Condición de Límite – Control de la Duración Absoluta de un Intento (*Limit Condition Attempt Absolute Duration Control*) que indica si el tiempo para un intento de una actividad ha sido establecido; y Condición de Límite – Duración Límite Absoluta para un Intento (*Limit Condition Attempt Absolute Duration Limit*) que indica en segundos disponibles para un intento de una actividad.

## Descripción de Reglas de Rollup

Las **Reglas de Rollup** es un mecanismo utilizado para llevar un control sobre la información del progreso del alumno en los Clusters debido a que no tienen asociados objetos de contenido y no existe otro mecanismo para tal fin. Estas reglas tienen el siguiente formato: if [conjunto\_condiciones] “cierto” for [conjunto\_actividades\_hijas] then [acción]. Para cada regla se evaluarán un conjunto de condiciones en función de la información de seguimiento almacenada para un conjunto de actividades hijas del Cluster durante el Proceso Global de Rollup para ejecutar una acción en caso de ser satisfactorias. De igual manera que en las Reglas de Secuencia, se pueden agrupar conjuntos de condiciones bajo las cláusulas “All” y “Any”, así como también pueden usarse los operadores unarios “Not” y “NO-OP”.

Las **condiciones** que pueden aparecer en estas reglas son el siguiente conjunto de las que ya vimos para Reglas de Secuencia: Satisfacción, Conocimiento del Estado del Objetivo, Conocimiento de la Medida del Objetivo, Completado, Conocimiento del Progreso de la Actividad, Intentado, Excedido el Límite de Intentos. El concepto es el mismo, la única diferencia es que aquí hablamos de objetivo de Rollup asociado a la Actividad de los hijos, ya que en estas reglas se evalúan Clusters.

A la hora de evaluar las Reglas de Rollup se tiene en cuenta la información de seguimiento almacenada para todas las Actividades hijas del Cluster, pero se pueden especificar **restricciones** sobre esta información:

- Se puede definir “Seguimiento” a falso, de manera que no se almacenará información ninguna sobre el estado del seguimiento.
- Se puede definir la “Satisfacción del Objetivo de Rollup” a falso por lo que no se incluirá en la evaluación de la Regla de Rollup la Actividad que tenga Satisfecha o no Satisfecha la Acción de Rollup.
- Se puede definir la “Medida del Peso del Objetivo de Rollup” a 0.0 indicando que la valoración de la Actividad no contribuirá a media del peso medido de su padre.

- Se puede definir la “Finalización del Progreso del Rollup” a falso por lo que no se incluirá en la evaluación de la Regla de Rollup la Actividad que tiene Completa o Incompleta la Acción de Rollup.
- Se puede definir la “Medida de la Satisfacción si Activa” que indica cuando la medida de un objetivo de Rollup de una Actividad puede ser incluido en la satisfacción de un objetivo de Rollup.
- Se puede definir los “Elementos Requeridos para Rollup” que indica de manera condicional cuando una Actividad puede ser incluida en la evaluación de Reglas de Rollup teniendo especificadas Acciones de Rollup.

El **Conjunto de Actividades Hijas del Rollup** define como los resultados de condiciones de las actividades incluidas en la evaluación determinan la ejecución o no de la Acción de Rollup. Tenemos los siguientes operadores para realizar esta determinación:

- **Todos** (*All*): para ejecutar la acción todas las actividades tienen que evaluar cierto sus condiciones.
- **Cualquiera** (*Any*): para ejecutar la acción cualquiera de las actividades tiene que evaluar a cierto sus condiciones.
- **Ninguno** (*None*): para ejecutar la acción ninguna de las actividades tiene que evaluar a cierto sus condiciones.
- **Por lo Menos Cuenta** (*At Least Count*): para ejecutar la acción un número determinado por “Mínima Cuenta de Rollup” de actividades tienen que evaluar a cierto sus condiciones.
- **Por lo Menos un Porcentaje** (*At Least Percent*): para ejecutar la acción un porcentaje determinado por “Mínimo Porcentaje de Rollup” de actividades tienen que evaluar a cierto sus condiciones.

Las **Acciones** de Rollup existentes que se pueden ejecutar durante el Comportamiento de Rollup y que pueden afectar al modelo de estado del seguimiento son las siguientes:

- **Satisfecho** (*Satisfied*): Pone a cierto el valor “Estado del Progreso del Objetivo” y “Estado de Satisfacción del Objetivo” para el objetivo rollup asociado a la actividad.
- **No Satisfecho** (*Not Satisfied*): Pone a cierto el valor “Estado del Progreso del Objetivo” y a falso el valor “Estado de Satisfacción del Objetivo” para el objetivo rollup asociado a la actividad.
- **Completado** (*Completed*): Pone a cierto el valor “Estado del Progreso del Intento” y “Estado de Finalización del Intento” para el objetivo rollup asociado a la actividad.
- **Incompleto** (*Incomplete*): Pone a cierto el valor “Estado del Progreso del Intento” y a falso el valor “Estado de Finalización del Intento” para el objetivo rollup asociado a la actividad.

Se pueden añadir restricciones condicionales a la contribución de las Actividades al Rollup de su padre con los **Controles de Rollup**. Estos son los tres tipos de información de seguimiento que se pueden restringir:

- **Satisfacción del Objetivo de Rollup** (*Rollup Objective Satisfied*): indica si contribuye o no a la evaluación de Satisfacción de la Regla de Rollup del padre de la Actividad.

- **Medida del Peso del Objetivo de Rollup** (*Rollup Objective Measure Weight*): Aplica un valor de peso (valor de tipo real) a la Medida Normalizada del Objetivo asociado a la Actividad durante el Rollup de la Actividad padre.
- **Finalización de Progreso de Rollup** (*Rollup Progress Completion*): indica si contribuye o no a la evaluación de Finalización de la Regla de Rollup del padre de la Actividad.

### Descripción de los Objetivos

Las Actividades del Árbol pueden llevar asociados uno o varios objetivos. Estos son los elementos que deben ser descritos para definir cada objetivo:

- **ID de Objetivo** (*Objective ID*): Identificador del Objetivo. Sirve para asociar la “Información de Progreso del Objetivo” con la actividad.
- **Satisfacción de un Objetivo por Medida** (*Objective Satisfied by Measure*): indica si la “Mínima Medida Normalizada de Satisfacción” puede ser usado para determinar si el objetivo ha sido satisfecho.
- **Mínima Medida Normalizada de Satisfacción** (*Objective Minimum Satisfied Normalized Measure*): si esta habilitado para ser usado (como indica el elemento anterior) indica la medida mínima para considerar que se ha cumplido el objetivo con un valor normalizado entre [-1.0...1.0]. En caso de estar habilitado y de que el “Estado de Medida del Objetivo” sea cierto podrá afectar al estado del objetivo. Rellenará como cierto los valores “Estado de Progreso del Objetivo” y “Estado de Satisfacción del Objetivo” si la “Medida Normalizada del Objetivo” es mayor que este valor. Si la “Medida Normalizada” es menor al valor presente en este elemento, rellenará como cierto “Estado de Progreso del Objetivo” y como falso “Estado de Satisfacción del Objetivo”. Por defecto el valor es 1.0.
- **El Objetivo Contribuye al Rollup** (*Objective Contributes to Rollup*): indica si el “Estado de Satisfacción” y la “Medida Normalizada” del Objetivo serán usadas durante el Rollup.

Los objetivos locales están asociados a actividades y no pueden ser consultados por otras actividades. Para conseguir objetivos comunes entre actividades encontramos los objetivos globales. Para ello se debe definir un **Mapa de Objetivos**. Mediante este Mapa se puede compartir además Información del Progreso de los Objetivos que aquí aparezcan. Los Árboles de Actividades tienen un elemento de tipo booleano que regula el alcance y la duración de los Objetivos Globales, conocido como “Objetivos Globales para el Sistema”. Si es cierto habilita la existencia de la información de los Objetivos Globales durante todo el tiempo que el alumno este en el sistema, sino solo para un Intento.

El Mapa de Objetivos tiene la siguiente información (todos los elementos aquí descritos son booleanos): “ID de Objetivo para Actividad” e “ID del Blanco del Objetivo”, que asocia un objetivo local a uno global para que compartan información; “Leer” y “Escribir Estado de Satisfacción de un Objetivo” permiten al objetivo global obtener y escribir información sobre los valores “Estado del Progreso” y “Estado de la Satisfacción de un Objetivo”; “Leer” y “Escribir Estado” permiten al objetivo global obtener y escribir información sobre los valores “Estado de la Medida” y “Medida Normalizada del Objetivo”.

## Controles de Entrega

Los Controles de Entrega sirven para ayudar al EVE a gestionar la información del estado del seguimiento. Así indicarán si un SCO asociado a una actividad puede comunicar distintos tipos de información de seguimiento. Encontramos en este tipo de control el elemento booleano “Rastreo” (“*Tracked*”) indica que pueden ser almacenadas y que contribuirán al Rollup de la actividad padre los valores “Información del Progreso del Objetivo” y “Información del Progreso de la Actividad/Intento” a menos que otra información de secuencia no indique lo contrario. El elemento “Finalización Fijada por Contenido” especifica si el SCO asociado a la actividad puede fijar el “Estado de Finalización de Intento”. El elemento “Objetivo Fijado por Contenido” especifica si el SCO asociado a la actividad puede fijar el “Estado de Satisfacción de Objetivo” del objetivo asociado a la actividad. Para poder realizar esta acción, además el objetivo debe tener a cierto el valor “Objetivo Contribuye a Rollup”.

## Otros Controles

SCORM permite a los desarrolladores añadir determinada información de secuencia para seleccionar un número determinado de actividades (**Controles de Selección**), modificar de forma aleatoria el orden de las actividades hijas en un Cluster (**Controles de Aleatoriedad**) y ayudar al EVE a gestionar la información del estado del seguimiento (**Controles de Entrega**).

Los Controles de Selección permiten elegir cuando (Nunca, Una Vez, Para Cada Nuevo Intento) se va a permitir realizar la selección mediante el elemento “**Timing de Selección**” (“*Selection Timing*”). Con los elementos “**Estado de la Cuenta de Selección**” (“*Selection Count Status*”) y “**Cuenta de Selección**” (“*Selection Count*”) se puede definir el número de actividades que van a poder ser seleccionadas entre las Actividades hijas de la Actividad para la que se asigna este control. El primer elemento es un booleano que indica que la “Cuenta de Selección” tiene significado para la Actividad. La “Cuenta de Selección” indica el número máximo de Actividades que se pueden seleccionar. Sólo se pueden aplicar este tipo de controles a Clusters.

Los Controles de Aleatoriedad permiten elegir cuando (Nunca, Una Vez, Para Cada Nuevo Intento) va a ser aleatorio el orden de las Actividades hijas mediante el elemento “**Timing de Aleatoriedad**” (“*Randomization Timing*”). El elemento “**Hijos Aleatorios**” (“*Randomize Children*”) es un booleano que indica si las Actividades hijas de la Actividad para la que se define este control se mostrarán en orden aleatorio. Sólo se pueden aplicar este tipo de controles a Clusters.

Los Controles de Entrega (*Delivery Controls*). Indican si un recurso (Asset, SCO) asociado a una Actividad puede comunicar distintos tipos de información de seguimiento. Encontramos en este tipo de control el elemento “**Rastreo**” (“*Tracked*”) {ambos;cierto} indica que pueden ser almacenados y que contribuirán al Rollup de la Actividad padre los valores “Información del Progreso del Objetivo” y “Información del Progreso de la Actividad/Intento” a menos que otra información de secuencia no indique lo contrario. El elemento “**Finalización Fijada por Contenido**” {hojas;falso} especifica si el recurso asociado a la Actividad puede fijar el “Estado de Finalización de Intento”. El elemento “**Objetivo Fijado por Contenido**” {hojas;falso} especifica si el recurso asociado a la Actividad puede fijar el “Estado de Satisfacción de Objetivo” del Objetivo asociado a la actividad. Para poder realizar esta acción, además el objetivo debe tener a cierto el valor

“Objetivo Contribuye a Rollup”. Los dos últimos campos vistos permiten gestionar la información de secuencia cuando el recurso asociado a la Actividad es un Asset, ya que este tipo de contenido no comunica nada al EVE. De esta manera si el primero de estos dos campos es falso se marcará la Actividad como “completada” a menos que el contenido reporte otro estado de completitud. Si el segundo campo es falso se marcará la Actividad como “satisfecha” si el contenido no almacena otro estado de satisfacción.

### Controles de Navegación

Permiten controlar como serán mostrados los elementos visuales que permiten navegar al alumno por los contenidos. Sólo pueden ser aplicados a las hojas del Árbol. El único tipo es **Ocultar Elementos del UI del LMS** (*Hide LMS UI Values*) que permite definir para cada actividad que elementos de Navegación serán mostrados. Sus campos son valores booleanos a falso por defecto. Cuando se activan, ocultan el elemento visual en cuestión. Los campos son “Previo” (oculta el botón que permite ir la Actividad anterior), “Continuar”, “Salir”, “Salir de Todos”, “Abandonar”, “Abandonar Todos” y “Suspender Todos”.

### **Proceso de Secuenciamiento**

El Proceso de Secuenciamiento abarca control global de los comportamientos de secuencia para la implementación dada por el EVE. De esta manera queda definido como se aplican los comportamientos de Secuenciamiento (comportamiento de Rollup, de finalización, de secuencia, etc) durante una sesión de secuencia. Una sesión de secuencia comprende cuando se lanza un curso, cuando el usuario hace una petición de Navegación o cuando un SCO termina.

Para describir el proceso se utiliza el concepto de **Loop de Secuenciamiento** para decidir cual será la siguiente Actividad a mostrar. En el Loop de Secuenciamiento se definen un conjunto de algoritmos que utilizan las reglas de secuencia y el estado de los datos de seguimiento para tomar ejecutar una Actividad en concreto. La especificación de los algoritmos de secuencia viene dada por un pseudocódigo que el EVE debe implementar para llevar a cabo el comportamiento explicado en SN. Analicemos el Proceso de Secuenciamiento desde un punto de vista conceptual:

- 1) Se recibe un evento de navegación, se comprueba su validez y se transforma a una petición de secuencia concreta.
- 2) Termina el Intento para la Actividad, actualizando el estado del Árbol de Actividades. Para ello se traslada la información recogida en el modelo de datos del RTE y se realizan las acciones de Rollup pertinentes.
- 3) Determina que actividades del Cluster pueden intervenir en el proceso de selección de la Actividad a ejecutar (siguiente paso).
- 4) Se comprueban las reglas de secuenciamiento para decidir cual será la siguiente Actividad a ejecutar.
- 5) Comprueba si sobre la Actividad seleccionada no hay ningún control que impida su ejecución, mostrándola en ese caso.

## Ejemplo ilustrativo

Veamos como se puede controlar la secuencia de actividades usando SCORM SN. Vamos a mostrar un ejemplo en el que en función de la actividad del alumno en una “Actividad1” se permitirá ejecutar o la “Actividad2” o la “Actividad3”.

Supongamos que en la “Actividad1” lanza un SCO que crea y modifica el estado de un objetivo en el modelo de datos del RTE. Para poder realizar una adaptación del contenido condicionaremos la posibilidad de que sean ejecutadas “Actividad2” y “Actividad3” en función de que la “Actividad1” consiga superar o no el objetivo. Supongamos que si se cumple el objetivo se ejecutará la “Actividad3”, y si no se cumple, se ejecutará la “Actividad2”

El campo “sucess\_status” del objetivo del modelo de datos RTE es el que se va a condicionar la bifurcación entre las dos actividades hijas. Este campo se relaciona con el “Estado de Progreso del Objetivo” para la “Actividad1” en el modelo de seguimiento de SN. El SCO de la “Actividad1” se encargará de poner el valor “passed” en el campo explicado del modelo de datos RTE cuando considere que el alumno completo el objetivo propuesto en la actividad.

Por tanto utilizando una regla de secuencia para cada una de las actividades de la bifurcación controlaremos su acceso. Para definir la regla de secuencia debemos especificar la condición que activará la regla y la acción asociada.

Para la “Actividad3” se debe definir una condición del tipo “Objective Status Known”, que será cierta si el campo “sucess\_status” es “passed”, y falso para otros valores. La acción asociada será “Disabled”, la cual impide que se ejecute la actividad. Para la “Actividad2” se define la misma regla añadiendo en la condición un operador de negación.

## 2.3 Entornos Virtuales de Enseñanza (EVE)

Encontramos el *blended learning* como una de las tendencias dentro del *e-Learning* con mayor aceptación dentro de la enseñanza superior universitaria. La idea de complementar la formación recibida en el aula con un conjunto de actividades que gracias a Internet pueden ser desarrolladas casi en cualquier lugar y momento ha contribuido a esta aceptación. La asistencia presencial no es sustituible por ninguna herramienta tecnológica pero si es interesante complementar la formación con el uso de los nuevos avances en el campo.

Aunque el uso de las nuevas tecnologías en la educación no es nada nuevo sino que data de varias décadas, el desarrollo de Internet y la digitalización de la información han contribuido a que se utilice un nuevo soporte para la transmisión del conocimiento (Adell 1997). De esta manera se encuentra los Entornos Virtuales de Enseñanza como un espacio virtual en el que poder desarrollar una experiencia educativa que complemente y refuerce la formación recibida en los centros de enseñanza. Y muestra de esta aceptación la encontramos en la existencia de “Campus Virtuales” en la práctica totalidad de universidades.

La comunidad universitaria europea en general y la española en particular se ven envueltas en un proceso de cambio tanto en las metodologías de aprendizaje como en los contenidos y estructura de las titulaciones. Este cambio viene motivado por la adhesión al marco del Espacio Europeo de Educación Superior (EEES) que pretende la creación de un entorno

de educación común para fomentar la movilidad de los estudiantes y del conocimiento. Una de las bases del EEES reside en dar mayor protagonismo al estudiante en el proceso formativo. Esta es una de las características del *e-Learning*, situar al estudiante en el centro del proceso de enseñanza-aprendizaje, por lo que utilizar una plataforma de *e-Learning* como parte del proceso parece una decisión acertada y acorde esta idea que presenta el EEES (Esteve 2009). De esta manera se pretende dotar al alumno de un conjunto de directrices y herramientas para que sea capaz de adquirir las destrezas necesarias para aprender por su cuenta: Es decir, “aprender a aprender” (Duart y Sangrà 2000).

Otro de los puntos fuertes en los que tanto el EEES como multitud de autores hacen hincapié es el de fomentar el aprendizaje colaborativo. Este tipo de aprendizaje se basa en unir los esfuerzos de un conjunto de personas con la intención de alcanzar una meta común. Difiere ligeramente del concepto de trabajo en grupo en cuanto que no consiste simplemente en sumar esfuerzos de los integrantes y la compartir información entre ellos, sino que deben cooperar para alcanzar el fin colectivo (Echazarreta et al 2009). Uno de los objetivos de la universidad es el de formar profesionales. Uno de los aspectos a los que se van a enfrentar estos “nuevos profesionales” una vez abandonen la universidad es el de trabajar en equipo, por lo que supone una ventaja extra el hecho de que tengan una experiencia mayor en ese aspecto. Esto queda reflejado en los EVE’s actuales ya que tienen multitud de herramientas que fomentan y soportan el aprendizaje colaborativo.

La introducción del aprendizaje colaborativo no es una tarea sencilla, y presenta ciertas barreras que deben superarse para que cale en el modelo universitario. Para ello es necesario introducir los nuevos avances en herramientas software que faciliten el trabajo en grupo junto con la evaluación de los aspectos socio-culturales que intervienen en este proceso (Barker 1999). Una de las mayores barreras es el rechazo a la utilización de estos métodos por parte de los alumnos, por no estar acostumbrados a estos métodos y por el miedo a la falta de organización en el grupo y al trabajo insuficiente de algunos de los miembros del mismo. Es decir, hay una carencia desde el punto de vista cultural y educativo. Por ello es necesaria la figura del profesor. Debe guiar y hacer conscientes a sus alumnos de la importancia y los beneficios del trabajo en grupo, y mostrarse como una figura mediadora durante todo el proceso. Con ello se consigue que los alumnos se sientan parte del proceso de aprendizaje en grupo y sean menos reticentes a la consecución de las prácticas que propone esta forma de trabajo (Hogarth 2008). Los EVE soportan distintos tipos de roles y brindan el soporte tecnológico necesario para facilitar este proceso tanto a profesores como alumnos. Pero no sirve para nada tener los medios tecnológicos si no existe una formación del profesorado tanto en estas nuevas técnicas de enseñanza-aprendizaje como en la utilización de estas nuevas tecnologías (creación de cursos virtuales, reutilización y creación de contenidos, etc.) (Echazarreta et al 2009).

Todo esto no hace más que dar una mayor importancia a la utilización de los Entornos Virtuales de Enseñanza, fomentando además los esfuerzos de trabajo tanto en la mejora de las herramientas y contenidos utilizados como la incorporación de nuevas utilidades.

### **2.3.1 Descripción**

Para entender correctamente el término de Entorno Virtual de Enseñanza (EVE) conviene que hagamos un pequeño análisis previo de las plataformas *e-Learning*. Una *plataforma de e-Learning* es un software que se encarga de la gestión de usuarios (alta, login), de contenidos (creación, ejecución, monitoreo, control de acceso), servicios de comunicación (foros, correo, chat) y que se encuentra alojado en un servidor para que puedan acceder a los

servicios, generalmente vía web. Distintos autores diferencian entre Sistema de Gestión de aprendizaje (Learning Management System) y Sistema de Gestión de Aprendizaje y Contenidos (Learning Content Management System) diferenciándolos en que el LCMS permite la autoría de contenidos y los LMS no. También diferencian entre LMS y EVE en que los EVE no permiten la utilización de contenidos que no hayan sido creados para esa plataforma en concreto. Hoy en día se pueden utilizar indistintamente los términos de LMS o EVE, y con el se abarca todo el rango de las plataformas de *e-Learning* (Paulsen 2002).

Por tanto, un Entorno Virtual de Enseñanza (*Virtual Learning Environment*) es una herramienta software cuyo cometido es el de gestionar la experiencia de aprendizaje electrónica. Automatiza la manera de dirigir el proceso de enseñanza. Existen multitud de sistemas EVE tanto de pago como de libre distribución, de los cuales podemos extraer el siguiente conjunto de funcionalidades que brindan:

- Gestión de Usuarios y Roles: permiten registrar usuarios, asignarlos distintos roles (profesor, alumno, administrador) y controlar su acceso al sistema, distinguiendo las posibles opciones de uso del mismo.
- Administración y control de cursos: el curso es la unidad básica de organización de los EVE's. Permite a los creadores de los cursos asignarles contenidos educativos, crear grupos de trabajo y realizar una planificación temporal de los mismos.
- Seguimiento y evaluación de la actividad del alumno: los EVE permiten realizar un seguimiento del proceso de aprendizaje del alumno y permiten realizar una valoración de su esfuerzo mediante la integración de determinadas herramientas de evaluación. Dos de las más soportadas en distintos EVE son los test on-line y la entrega de trabajos.
- Comunicación entre Usuarios: brindan un conjunto de herramientas de distinta índole para favorecer la comunicación entre los miembros del sistema. Varían en función del EVE, encontrando foros, chats, videoconferencias, correo electrónico, etc.

En la actualidad encontramos multitud de EVE de libre distribución y de pago tanto en organizaciones privadas como públicas.

## LAMS

LAMS<sup>11</sup> (*Learning Activity Management System*) es una herramienta EVE de libre distribución desarrollada en colaboración por LAMS Foundation Ltd, LAMS International Pty Ltd y Macquarie E-learning Centre Of Excellence bajo el amparo de la universidad Macquarie University de Sydney, Australia. LAMS permite el diseño de contenidos educativos colaborativos mediante la creación de secuencias de actividades basándose en la especificación Learning Design de IMS (Dalziel 2003). Una vez creadas, LAMS da soporte a la ejecución y seguimiento de la interacción del usuario con la actividad.

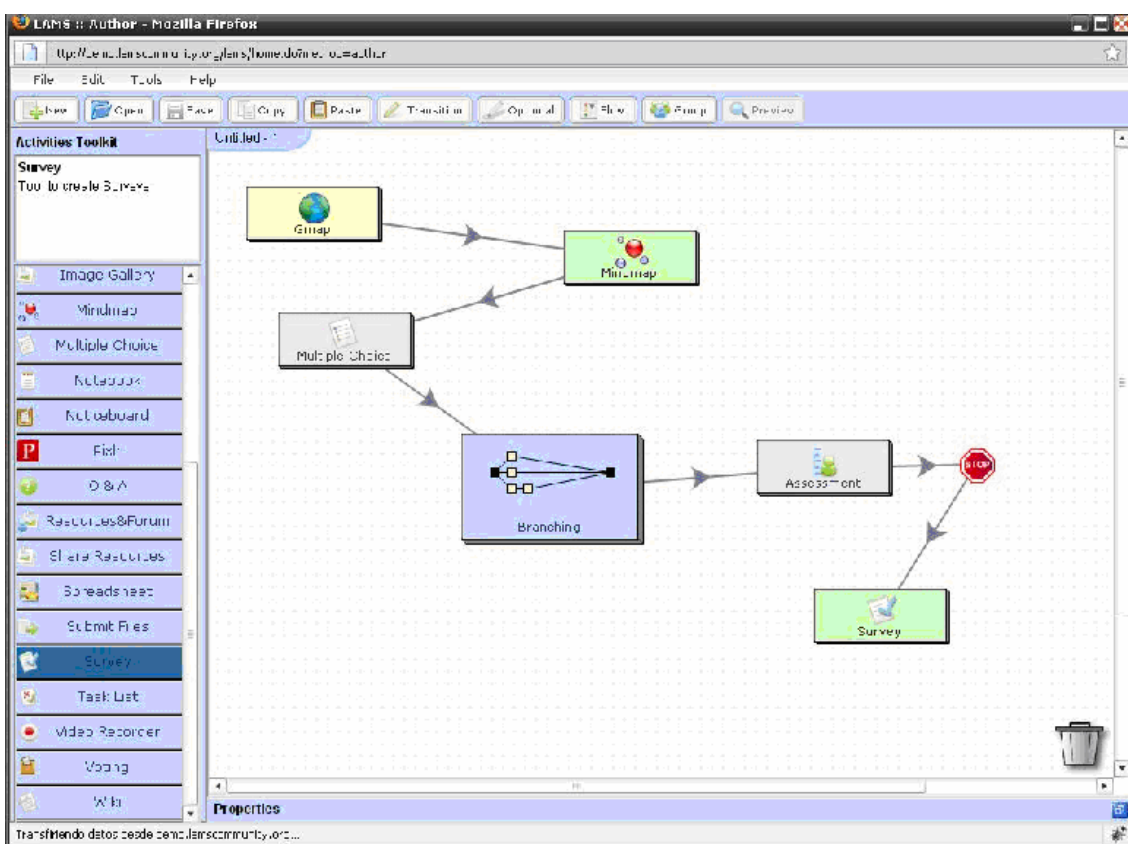
LAMS tiene distintos entornos para llevar a cabo las tareas comentadas. En el entorno de autoría permite crear una secuencia de actividades de una manera visual. Estas actividades incluyen herramientas de comunicación como chat, foros, libretas en las que anotar las opiniones del alumno, encuestas, videoconferencias, etc.; herramientas de evaluación como cuestionarios de respuesta múltiple, de pregunta/respuesta o definidas por el tutor; así como multitud de actividades para añadir contenidos externos, desde subir un OA hasta fotos, presentaciones, imágenes, videos, mapas conceptuales, etc. Una de las ventajas de

---

<sup>11</sup> <http://wiki.lamsfoundation.org>

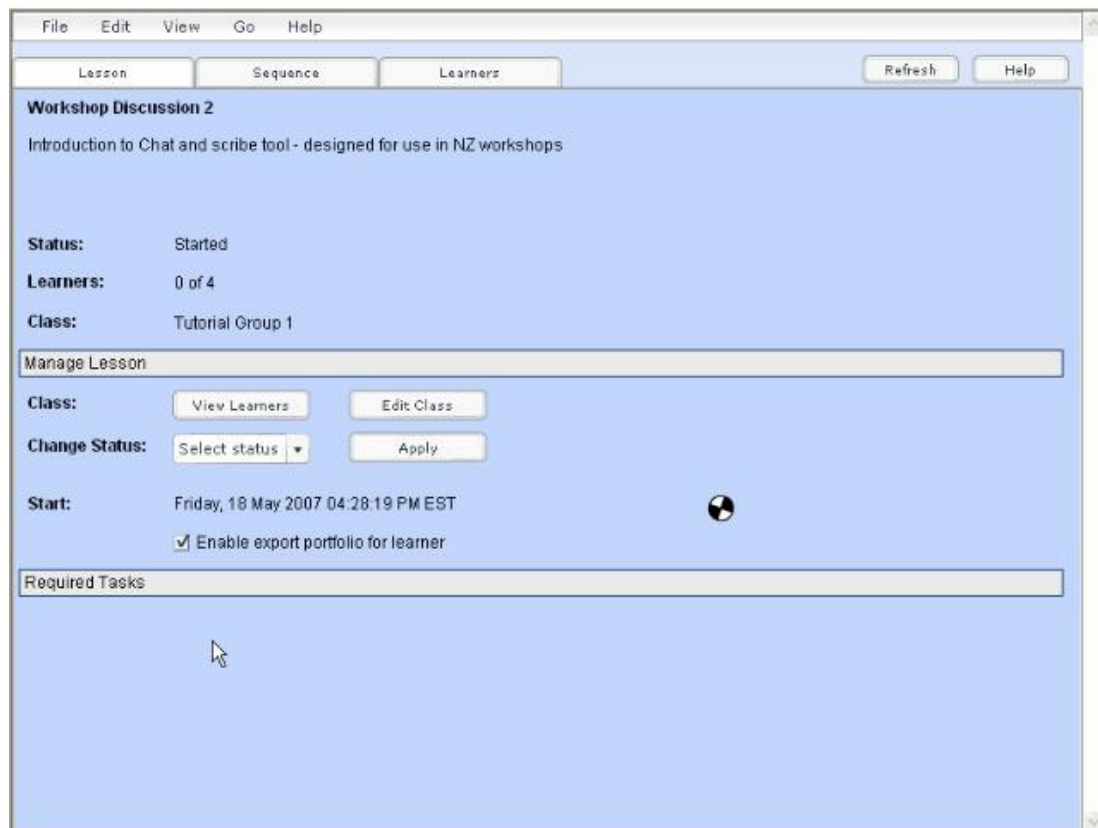
LAMS es que, a parte de soportar actividades individuales, permite actividades que fomentan un alto nivel de participación del alumnado, de forma que el tutor puede presentar una actividad y que esta vaya siendo completada por medio de la participación de todos los alumnos de un curso, o por los alumnos de un grupo.

LAMS permite controlar el flujo de las actividades en una secuencia mediante el uso de unos elementos especiales: las “gates” (puertas) que permiten detener la secuencia en un determinado punto hasta que se cumplan una serie de propiedades como que todos los alumnos hayan participado en una actividad, que el autor de la secuencia lo permita a través del monitoreo de la secuencia, o que se hayan cumplido una serie de condiciones relacionadas con la actividad del alumno en actividades anteriores; el “branching” (ramificación) que podrá definir distintos caminos en la secuencia en función de la interacción del alumno en actividades pasadas; o definir actividades opcionales, en las que se permite elegir al usuario entre un conjunto de actividades.



**Figura 3: Entorno de autoría: Se permite crear secuencias de contenidos para ser ejecutados en LAMS.**

Otro aspecto interesante en LAMS es el entorno de monitoreo, en el cual se permite al tutor llevar un control de las interacciones de los alumnos en tiempo real, permitiendo cambiar el estado de una secuencia durante su ejecución para activarla, suspenderla o borrarla, añadir puertas o ramificaciones o que cambie el contenido de las mismas.



**Figura 4: Entorno de monitorización de LAMS. Permite realizar un seguimiento de la actividad del alumno en la secuencia, y modificar su flujo.**

LAMS se presenta como un EVE en sí, permitiendo la crear, compartir, reutilizar, ejecutar y monitorizar secuencias de contenidos colaborativos. Pero además soporta su integración en multitud de EVE. De esta manera, podemos añadir un módulo en un EVE en concreto para conseguir la comunicación entre los dos sistemas, de manera que sirva como nexo para conseguir que se comporte como un solo sistema sin tener que realizar el acceso en ambos, con realizar el registro en el EVE en el que está integrado LAMS es suficiente. Las secuencias creadas en LAMS podrán utilizarse así en un curso del EVE en el que está integrado. LAMS puede integrarse en Moodle, Sakai, Blackboard, .LRN y muchos EVE más.

## Moodle

Moodle<sup>12</sup> se presenta como un EVE de libre distribución creado para facilitar a los educadores a la creación de colectivos de de aprendizaje en la red. Esta basado en tecnología LAMP (Linux Apache MySQL y PHP). Surge de la iniciativa de Martin Dougiamas a través de la experiencia adquirida como administrador de WebCT en la Universidad Tecnológica de Curtin. Su primera versión se liberó en 2002, realizando nuevas versiones con regularidad. Tiene una amplia comunidad de usuarios y está siendo implantado en multitud de universidades como campus virtual.

El diseño de Moodle está ligado a las ideas constructivistas en pedagogía cuya principal idea se basa en la construcción del conocimiento por el estudiante en lugar de una simple transmisión del mismo, haciendo hincapié en que los alumnos pueden contribuir en esta construcción. Contiene herramientas que favorecen este tipo de aprendizaje, como foros,

<sup>12</sup> <http://moodle.org/>

wikis, encuestas, tareas compartidas, etc. Las interacciones de los distintos usuarios en estas herramientas dan la idea de comunidad a los cursos. Además permite importar recursos de muy distinta índole: contenidos html, audio, video, recursos IMS CP, SCORM, LAMS, etc.

Permite la creación de cursos en los cuales se pueden utilizar distintos modos de aprendizaje por lo que no obliga a dar un enfoque constructivista de la enseñanza. Permite la creación de contenido web o de evaluación. Moodle plantea la idea de curso como centro de la gestión de la tarea del aprendizaje, permitiendo varios formatos (semanal, por temas, módulos, etc.).



**Figura 5. Ejemplo de utilización de un Moodle para conocer sus recursos y herramientas**  
(<http://www.iescarrenomiranda.com/moodle/course/view.php?id=35>).

Moodle permite realizar calificaciones para los foros, diarios, tareas, cuestionarios, etc. Se permite al tutor hacer un seguimiento de la evaluación de cada alumno. Además permite hacer un seguimiento de la actividad del alumno en el curso, detallando la participación en foros, visitas a los módulos, entradas al curso y tiempo empleado, etc.

## Sakai

El proyecto Sakai<sup>13</sup> se presenta como una comunidad de instituciones públicas y privadas que han desarrollado un “Entorno de Colaboración y Aprendizaje” (Collaboration and Learning Environment, CLE) de libre distribución. Fue una iniciativa de la “Mellon Foundation”. Entre las instituciones que colaboran en su ciclo de vida se encuentran las universidades de Indiana, Michigan, Stanford y el MIT. Tiene un diseño modular para fomentar que instituciones y particulares colaboren en la integración de distintas funcionalidades del *e-Learning* en el sistema. Una de sus motivaciones es la de integrar un EVE en un portal institucional.

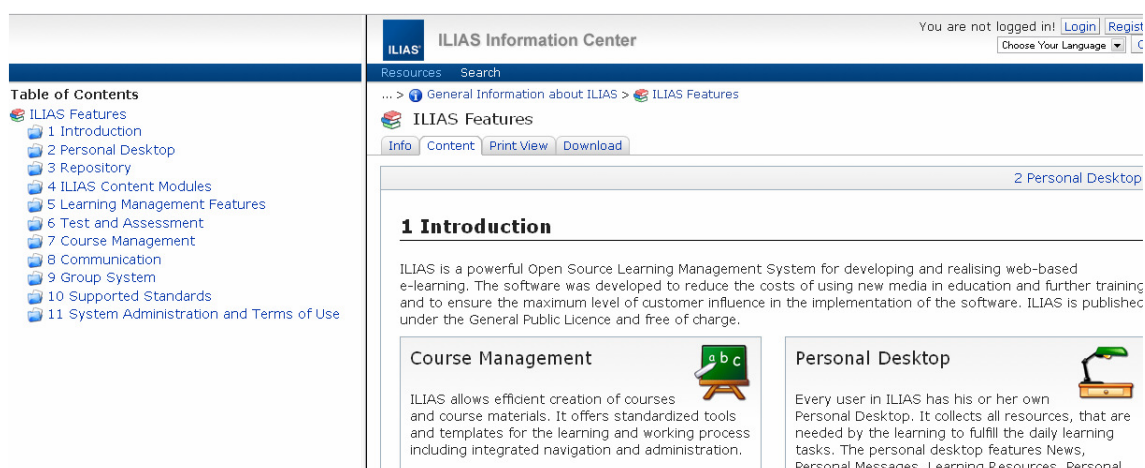
Sakai provee un entorno para la creación de un entorno de trabajo que se amolde a las especificaciones de cada curso, proyecto o investigaciones científicas. Sakai pretende dar soporte a los distintos tipos de aprendizaje en los cursos mediante un alto nivel de personalización del entorno basada en su modularidad. Así permite seleccionar a los tutores el conjunto de herramientas que se adecuen al modelo de aprendizaje que deseen utilizar. Además de herramientas de administración y evaluación, Sakai también da una importancia notable a las herramientas de tipo colaborativo. Para proyectos y/o investigaciones científicas favorecen la comunicación y el trabajo colaborativo.

<sup>13</sup> <http://sakaiproject.org/portal>

## Ilias

Ilias <sup>14</sup> es un EVE de código libre desarrollado por la Universidad de Colonia, Alemania, en colaboración con otras universidades europeas. Su motivación es la de proveer un EVE de bajo coste para facilitar la introducción de las nuevas tecnologías en la enseñanza teniendo en cuenta la aportación de los usuarios de este tipo de sistemas.

Permite la gestión de cursos y la personalización de la interfaz y suministra un conjunto de herramientas de comunicación, evaluación y permite utilizar distintos tipos de recursos. Cabe destacar que Ilias es el único EVE que tiene el certificado de compatibilidad con SCROM de ADL para las versiones 1.2 y 2004. Además posee un repositorio de los contenidos utilizados en el EVE.



**Figura 6: Ejemplo de utilización de un curso ILIAS para mostrar sus características ([http://www.ilias.de/docu/goto.php?target=lm\\_392&client\\_id=docu](http://www.ilias.de/docu/goto.php?target=lm_392&client_id=docu)).**

## WebCT y Blackboard

WebCT (*Web Course Tools*, o Herramientas para Cursos Web) es un EVE comercial por la iniciativa de Murray Goldberg, un miembro de la Universidad de Columbia Británica, en Canadá. Basa su éxito la flexibilidad de las herramientas utilizadas para la creación de los cursos, entre las que se encuentran las típicas utilizadas en este tipo de entornos.

Blackboard Inc. es una compañía de software Americana fundada en 1997. Blackboard se formó como firma consultora con un contrato con IMS Global Learning Consortium.

Blackboard Academic Suite es una suite de herramientas para la gestión del aprendizaje en línea. Incluye Blackboard Learning System que gestiona los cursos, Blackboard Community System que gestiona comunidades en línea y portales y Blackboard Content System, que gestiona el contenido del sistema. Por tanto, el funcionamiento de la suite es el que brindan los EVE.

Como ejemplo de la implantación de Blackboard encontramos sus productos presentes en más de 2200 instituciones educativas en más de 60 países.

Ambos sistemas se fusionaron bajo el nombre de Blackboard en el año 2006.

<sup>14</sup> <http://www.ilias.de/>

## 2.4 Conclusiones del Capítulo

El desarrollo del *e-Learning* esta ampliamente ligado al desarrollo de Internet y a la aceptación de este tipo de aprendizaje por multitud de instituciones. Para ser ampliamente aceptado diversas organizaciones ha trabajado para definir estándares que faciliten la interoperabilidad de contenidos. Los EVE han incrementado en número y en funcionalidades para conseguir integrar en el flujo educativo los contenidos electrónicos.

A pesar de que la iniciativa SCORM de ADL aparece con fuerza como un perfil de aplicación que aúna diversos estándares que abarcan varios aspectos (empaquetamiento de contenidos, comunicación contenido-EVE, secuencia de contenidos, etc.), hoy en día no hay ningún estándar ampliamente aceptado. Aunque el objetivo principal de los estándares es el de favorecer el intercambio y utilización de contenidos en cualquier sistema, el hecho de que no haya ninguno ampliamente aceptado dificulta las tareas de integración de contenidos en EVE. Para un mismo contenido deberemos crear múltiples versiones que se amolden a los distintos EVE que se adhieran a distintos estándares.

Esto unido a la diversidad de juegos existentes es un problema que dificulta la integración de videojuegos en EVE. Si además se desean explora el potencial adaptativo y de evaluación que presentan los juegos, el problema crece más aún.

En este trabajo presentamos una arquitectura que afronte tanto la diversidad de juegos como de estándares con el propósito de alejar a los desarrolladores de este tipo de contenido de las dificultades técnicas que supone. Esta arquitectura tiene en mente la adaptación y evaluación de juegos, por lo que ayudará a realizar dichas tareas.

### 3. Arquitectura

Actualmente existen distintos estándares para integrar y ejecutar contenidos educativos en servidores de aprendizaje. Ninguno de ellos tiene una aceptación lo suficientemente amplia para destacarlo sobre el resto. Esto afecta a la interoperabilidad de los contenidos, ya que hay que amoldarlos a las particularidades de cada estándar. A este problema hay que añadir la dificultad derivada de la complejidad de los estándares.

Cuando el contenido que se desea incorporar a un EVE es un juego, el problema crece aún más si cabe. Los juegos se presentan como contenidos altamente interactivos. Esta alta interactividad permite que entren en juego más factores a la hora de realizar tareas de adaptación, así como realizar una evaluación más completa de la actividad del alumno. Además para poder definir acciones de evaluación y adaptación en juegos, el desarrollador de este tipo de contenidos debe tener un conocimiento elevado tanto de los estándares como del juego en sí.

A continuación se describe una arquitectura general para la integración de videojuegos educativos en EVE de forma que se disminuyan los problemas tecnológicos asociados a ajustarse a los estándares y especificaciones de *e-Learning*. De esta manera se pretende aliviar el proceso de integración de estos contenidos en los servidores educativos y poder facilitar y mejorar los procesos de evaluación y adaptación en juegos.

#### 3.1 Visión general de la arquitectura

La arquitectura pretende resolver el problema de conectividad entre videojuegos educativos y distintos servidores EVE que cumplan con diversos estándares de *e-Learning*, ofreciendo un conjunto de servicios que posibiliten y mejoren las tareas de evaluación y adaptación de la experiencia educativa basada en juegos.

Planteamos la arquitectura subdividida en dos capas puesto que encontramos dos frentes abiertos a la hora de favorecer la interoperabilidad: el juego y el servidor. Por un lado hay que solucionar el problema de la comunicación con los servidores educativos. El desarrollo de juegos debe ser en la mayor medida posible independiente de los estándares o tipos de EVE con los que se va a comunicar. Por el otro, es necesario que las tareas de adaptación y evaluación puedan realizarse indistintamente del juego a utilizar. Además, parece interesante alejar a los creadores de contenidos del conocimiento de los estándares en términos de adaptación y evaluación para facilitar este proceso en el juego.

La **Capa de Comunicación (CC)**, que es la responsable de establecer y gestionar el canal de comunicación entre el EVE y el juego de manera estandarizada. Esta capa se encarga de acciones tales como “iniciar/finalizar comunicación” y “enviar/recibir datos”. El resultado de estas acciones depende del estándar a utilizar, siendo la CC la responsable de identificarlo. La CC tiene un conjunto de módulos (uno por estándar que implemente) que le brindan la información necesaria para poder realizar la comunicación con el EVE. Estos implementan una API común de manera que recurrirá a al módulo específico de un estándar cuando se intente realizar alguna de las operaciones comentadas.

La segunda capa es la **Capa de Adaptación y Evaluación (CAE)**, cuyo cometido es el de transformar la información recibida de la capa vecina en cambios concretos en el juego, y recopilar determinada información de la interacción del alumno en el juego para procesarla

y enviarla al EVE usando la CC. La CAE realiza peticiones de determinada información del alumno (progreso en la experiencia educativa, factores socio-culturales, etc.) que esta almacenada en el EVE. Además monitoriza la actividad del alumno en el juego. Puede determinar un conjunto de situaciones específicas que requieran adaptación en el juego mediante el procesamiento de la información recibida de ambas fuentes. Por ejemplo, que el alumno ya haya completado parte del juego, o que demostró conocimientos específicos planteados en el juego durante la consecución de otras actividades, por lo que cambiará el contenido que se presente, es decir, realizará una adaptación del mismo de forma transparente al usuario. La CAE puede realizar también tareas de evaluación. Utiliza la información recogida de la interacción del alumno para procesarla e identificar situaciones relevantes para la evaluación. Por ejemplo, que haya completado la mitad del juego, registrar una respuesta errónea a una pregunta, etc. Tanto las situaciones que deben producir evaluación como adaptación en el juego son definidas por el creador del contenido. Como ambas tareas son específicas, esta capa se descompone en otras dos subcapas para abordarlas de manera individual. Son la **Subcapa de Adaptación (SCA)** y la **Subcapa de Evaluación (SCE)**.

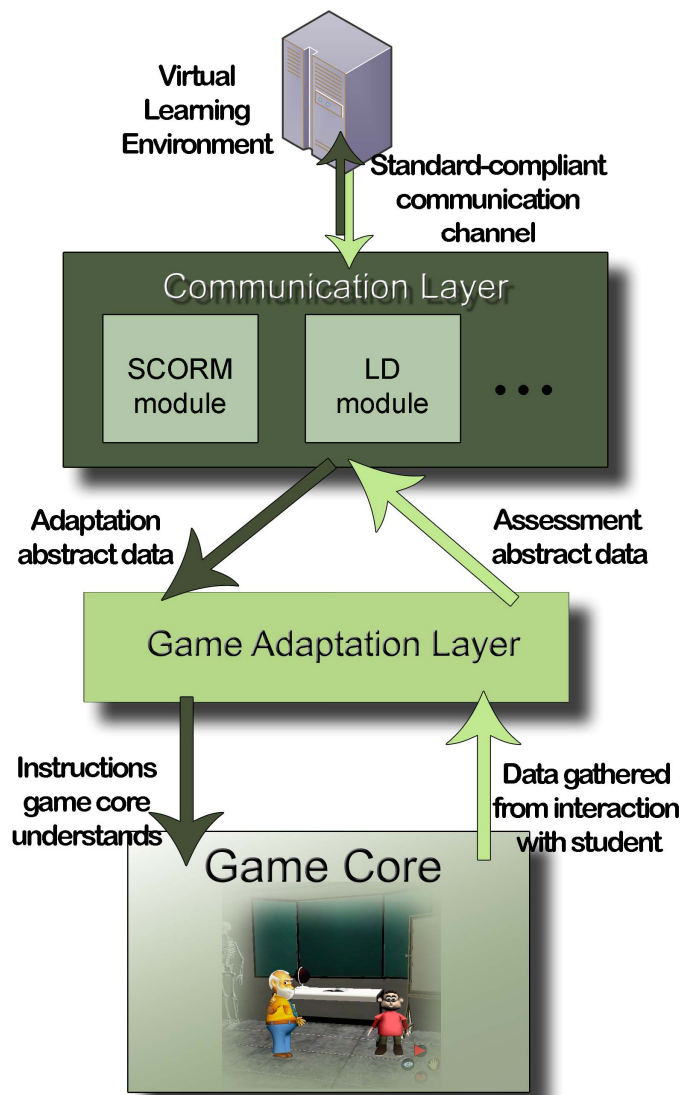


Figura 7: Visión general de la arquitectura. En esta figura se observan las dos capas. Los datos abstractos se recogen del Modelo de Datos Abstracto de Adaptación y Evaluación.

Además de las capas, la arquitectura incluye otro elemento: **el Modelo de Datos Abstractos de Adaptación y Evaluación** (MDAAE). Con este modelo se persigue incrementar la interoperabilidad de juegos y servidores educativos mediante un conjunto de términos abstractos que identifiquen y definan situaciones concretas que pueden darse entre los juegos y los EVE independientemente de su naturaleza. Estas situaciones a las que nos referimos incluyen tareas tanto de comunicación como de adaptación y evaluación. La CC facilita una API con métodos de comunicación en términos del MDAAE, como por ejemplo *recibir nombre del alumno* o *almacenar valoración de la actividad*. En el modelo se definen las situaciones en las que se deben ejecutar las tareas adaptación y evaluación y cuales serán dichas tareas. La CAE utilizará estos datos junto con la información recogida del juego y del EVE para procesar si se deben realizar o no y cual será la acción concreta a efectuar. Un ejemplo de tarea de evaluación, la capa detecta una situación de *respuesta correcta* y realizará la acción *aumentar en 1 la puntuación del alumno*.

### 3.2 El Modelo de Datos Abstractos

Como ya hemos visto, uno de los principales objetivos de la arquitectura es conseguir integrar videojuegos educativos en EVE independientemente del tipo de cada uno de ellos, es decir, del tipo de juego y estándar que utilice el servidor. De aquí viene que se utilicen términos abstractos de adaptación y evaluación entre las dos capas. Conociendo el significado de estos términos abstractos y dándoles soporte en el juego conseguiremos que tanto el EVE como el videojuego puedan ser independientes e intercambiables.

El objetivo de los juegos en términos de evaluación es la de proveer información sobre el rendimiento del jugador. La arquitectura middleware utiliza esta información junto con la que proporciona el perfil del estudiante almacenado en el EVE para llevar a cabo un proceso de adaptación del juego. Debido a alta interactividad permitida en los juegos, la adaptación puede llegar a ser más refinada que con otros contenidos educativos. Además la información recopilada también se puede utilizar para producir un informe de evaluación e incluirlo en el EVE y/o modificar el perfil del estudiante. Estas actualizaciones son utilizadas para futuras adaptaciones de la experiencia educativa, tanto en el juego como en otras actividades.

Para expresar qué acciones de adaptación deben realizarse cuando se detectan determinadas situaciones en el servidor educativo definimos el concepto de **regla de adaptación**. Las reglas de adaptación deben venir definidas de la forma que sigue:

**Si (estadoEVE) entonces (cambiarJuego)**

Donde “*estadoEVE*” es un término abstracto de adaptación que define un conjunto de condiciones que se deben dar en el modelo de datos descrito para el EVE, y donde “*cambiarJuego*” es un término abstracto que identifica un conjunto de acciones que se deberán ejecutar en el juego.

De la misma manera, debemos definir el concepto de **regla de evaluación** para reflejar qué acciones deben realizarse en el estado del servidor educativo cuando se detectan determinadas circunstancias en el juego derivadas de la interacción del alumno. Las reglas de evaluación están definidas como sigue:

**Si (estadoJuego) entonces (cambiarEVE)**

Donde “estadoJuego” es término abstracto de evaluación que identifica un estado concreto que debe darse para ejecutar la regla, y “cambiarEVE” que es un término abstracto que caracteriza como debe afectar al modelo de datos la situación detectada.

Este modelo debe definir un lenguaje de especificación para definir términos abstractos que identifiquen como adaptar y evaluar el juego de forma que se alejen las dificultades técnicas que esto supone. Además define la API de la CC en términos abstractos para que puedan ser utilizadas por futuros desarrolladores de la CAE para implementar los cambios concretos en el juego y en el servidor sin tener que adquirir las destrezas necesarias para afrontar las dificultades técnicas que supone. La implementación de este modelo debe almacenar los términos abstractos definidos en las siguientes categorías:

**Tabla 2: Categorías en las que se pueden dividir los términos abstractos.**

Categoría	Descripción
comunicacionAPI	Define las acciones de solicitar y almacenar información en el EVE.
estadoEVE	Define términos que identifican situaciones concretas del modelo de datos del EVE.
estadoJuego	Define términos que identifican estados específicos del juego
cambiarJuego	Define términos que identifican estados concretos del juego
cambiarEVE	Define términos que identifican estados concretos del modelo de datos del EVE.

Veamos un ejemplo para clarificar las ideas expresadas. Supongamos un término de la categoría comunicacionAPI cuyo significado es el de “obtener nivel de destreza del alumno”, la CC debe entender el significado que tendrá para cada uno los estándares que implementa. Por ejemplo en SCORM 2004 se concretaría en una llamada “GetValue(“cmi.score.raw”)”. De igual manera ocurre con el resto de términos. Los de la categoría estadoJuego deben ser interpretados por la SCE para definir el estado de juego a cada término y los de la categoría cambiarJuego deben ser interpretados por la SCA para definir un conjunto de cambios en el estado de juego. La capa de comunicación será la responsable, con la implementación de cada módulo, de interpretar los términos de las categorías comunicacionAPI, estadoEVE y cambiarEVE.

### 3.3 La Capa de Comunicación

La capa de comunicación se encarga de solicitar y transmitir información en ambas direcciones, hacia el EVE y hacia la otra capa. Abstrae la comunicación entre el juego y el servidor, aunque internamente depende del estándar que este implemente. Solicitará datos del perfil del estudiante recibiendo dicha información en el formato específico que brinde el estándar o la especificación y lo convertirá a términos abstractos de adaptación que entiende la capa correspondiente. A su vez, recibirá información de la CAE en términos abstractos de evaluación y los mandará para ser almacenados al EVE siguiendo la especificación que este utilice. Mediante el uso de la CC se consigue tanto recibir como almacenar información en el EVE con propósitos adaptativos y de evaluación independientemente del estándar que el servidor utilice.

En este punto nos surgió una duda importante. ¿Cómo sabe la CC que estándar utiliza el EVE? ¿Debe esta capa ir probando uno por uno cada estándar que tenga implementado hasta encontrar la respuesta? Esta solución no parece deseable.

La solución que proponemos se basa en un **Perfil de Ajustes de Comunicación**. Debe ser rellenado por el desarrollador del juego para elegir en cada caso particular que estándar o especificación utiliza el EVE en donde va a ser integrado. Además no todos los estándares tienen un modelo de datos ni un modelo de comunicación definido (por ejemplo, Learning Design (IMS Global Consortium 2003)). En este caso, el perfil debe contener la definición del protocolo de comunicación y/o el modelo de datos a utilizar y su relación con el modelo abstracto.

Como este perfil tiene la función de proveer a la capa de los datos necesarios para que pueda realizar la comunicación con el EVE, debe tener los campos expresados en la Tabla 3.

**Tabla 3: Campos del modelo de datos**

<b>Nombre del Campo</b>	<b>Tipo del Campo</b>	<b>Propósito</b>
<b>Tipo de Estándar/EVE</b>	<b>Cadena de Caracteres</b>	<b>Indica el estándar que debe usar o el EVE con el que la CC se va a comunicar. Indicará cual de los módulos de comunicación implementados va a ser usado.</b>
<b>Tiene Modelo de Datos</b>	<b>Booleano</b>	<b>Indica si el estándar a utilizar fijado en el campo “Tipo de Estándar/EVE” posee o no modelo de datos propio. En caso negativo utilizará el que se expone en el campo “Modelo de Datos”.</b>
<b>Tiene Protocolo</b>	<b>Booleano</b>	<b>Indica si el estándar a utilizar fijado en el campo “Tipo de Estándar/EVE” posee o no protocolo de comunicación propio. En caso negativo utilizará el que se expone en el campo “Protocolo”.</b>
<b>Modelo de Datos</b>	<b>Tipo Modelo</b>	<b>Modelo de Datos que será utilizado en caso de que el estándar no posea uno propio. El tipo Modelo constará de una tabla en la que se almacenan pares variable/valor..</b>
<b>Protocolo</b>	<b>Tipo Protocolo</b>	<b>Protocolo de comunicación que será utilizado en caso de que el estándar no posea uno propio. El tipo Protocolo especifica alguno de los protocolos de comunicación existentes (AJAX, REST, etc.), para el cual debe haber un módulo de comunicación en la CC.</b>
<b>Relación con el MDAAE</b>	<b>Tipo Relación</b>	<b>Almacena la relación entre el protocolo y modelo de datos definido y el MDAAE. Asocia términos abstractos a pares función del API - parámetros (campos del modelo de datos).</b>

Es importante resaltar que si se utilizan los campos “Modelo de Datos” y “Protocolo” debe añadirse un plug-in en el EVE con el que se pretenda comunicar, ya que no tiene porque entender los datos que le sean enviados.

### 3.4 La Capa de Adaptación y Evaluación

La CAE es la encargada de llevar a cabo las tareas de adaptación y evaluación en el juego con los datos solicitados al EVE, monitorizando la actividad en él y realizando los cambios pertinentes. Es por ello que esta capa está a su vez subdividida en otras dos capas, centrándose cada una de ellas en una tarea en concreto: la **Subcapa de Adaptación** (SCA) y la **Subcapa de Evaluación** (SCE).

La Capa de Comunicación envía la información recibida del servidor educativo a la SCA en términos abstractos de adaptación. Esta subcapa será la responsable de evaluar si con la información recibida se deben realizar o no cambios en el juego, llevándolos a cabo en caso afirmativo. La SCE se encarga de registrar la interacción producida en el juego y evaluar si las acciones realizadas deben ser almacenadas en el perfil del estudiante para realizar posteriores informes de evaluación, en cuyo caso notificará a la CC con instrucciones abstractas para que lo codifique y almacene en el servidor.

Con la descripción dada hasta el momento nos quedan algunos huecos por rellenar: ¿Cuáles son los criterios que utilizan las subcapas para elegir si adaptar el videojuego o valorar la actividad del alumno?

La Subcapa de Adaptación utiliza un **Perfil de Adaptación** en el cual se especifican las reglas de adaptación comentadas en la sección 3.2. Estas reglas contienen condiciones en el estado de juego y del EVE y unas acciones asociadas a su cumplimiento. Estas acciones quedan descritas en términos abstractos en el modelo de datos abstracto. La Subcapa de Evaluación tiene también un **Perfil de Evaluación** que sigue la misma idea que el perfil anterior. Define unas reglas basadas en el estado del EVE con unas acciones asociadas. Es responsabilidad de los desarrolladores del juego rellenar las reglas de ambos perfiles, tanto las condiciones como las acciones asociadas.

Los términos abstractos están definidos en el modelo de datos abstracto. Son las subcapas SCA y SCE las que deben “entender” el significado de cada uno de los términos para materializarlos en acciones concretas en el juego.

### 3.5 Conclusiones del Capítulo

Una de las mayores dificultades a la hora de incorporar videojuegos educativos en entornos *e-Learning* es la de mantener la interoperabilidad de juegos y de servidores educativos. Conseguir la interoperabilidad en este ámbito conlleva determinadas dificultades técnicas que recaen en la diversidad de estándares y juegos. Además, la dificultad aumenta si se pretende explotar el potencial que brindan los juegos en cuanto a adaptación y evaluación del contenido.

En este capítulo se da una aproximación a la solución del problema mediante una arquitectura que se sitúa entre las partes que se van a interconectar para aliviar las dificultades técnicas. La idea principal es abstraer las tareas técnicas de adaptación,

evaluación y comunicación entre el juego y EVE del proceso de creación de videojuegos educativos.

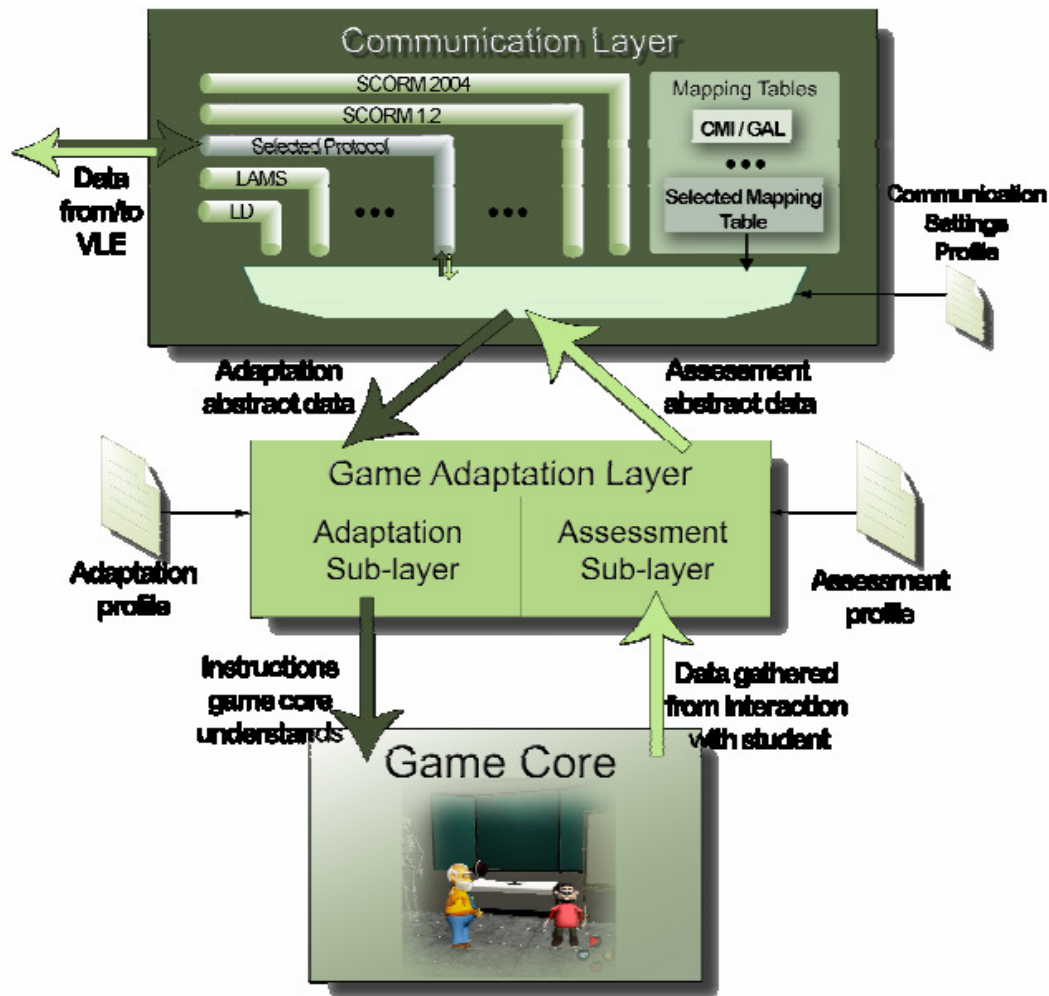


Figura 8: Visión de la arquitectura en la que entran en juego todos sus elementos (las dos capas y el modelo de datos). Además se pueden observar los ficheros que configuran la arquitectura.

Una vez planteado el modelo teórico deben trasladarse las ideas a un entorno real para comprobar en que medida la propuesta funciona. Para ello vamos a implementar la arquitectura en un determinado tipo de juegos para posteriormente integrarlos en distintos EVE.



## 4. Integración con <e-Adventure>

En el capítulo anterior se presenta una posible solución a los problemas derivados de la diversidad de estándares que se encuentran a la hora de integrar videojuegos educativos en servidores de aprendizaje.

Consideramos necesario, como parte de nuestra propuesta, demostrar que las ideas planteadas como solución realmente pueden incluirse en exportaciones de juegos para que puedan integrarse en EVE y llevar a cabo las tareas de adaptación y evaluación.

Para comprobar que la propuesta realizada puede llevarse a cabo, hemos implementado la arquitectura middleware multicapa en el entorno de autoría de creación de videojuegos educativos <e-Adventure>.

### 4.1 La plataforma <e-Adventure>

<e-Adventure> (Moreno-Ger et al 2008, Torrente et al 2008) es una plataforma creada con el propósito de facilitar la introducción de juegos y simulaciones basadas en juegos en el proceso de enseñanza. La herramienta consta de un editor gráfico para la creación de los juegos y un motor que permite ejecutarlos. Pretende acercar el desarrollo de videojuegos a los profesores para que puedan complementar con este tipo de actividades las enseñanzas que imparten. Además, el hecho de que profesionales de la enseñanza participen activamente en la creación de juegos conlleva que estos posean un valor educativo mayor. Para ello <e-Adventure> intenta superar dos de las barreras que dificultan que participen activamente en este proceso: el coste de desarrollo y las dificultades técnicas. La plataforma se centra en la producción de aventuras gráficas, del tipo *MonkeyIsland*© o *Myst*©, ya que este tipo de juegos es adecuado para la transmisión de conocimientos por sus características narrativas y por que su desarrollo se puede realizar a muy bajo coste. <e-Adventure> añade determinadas elementos específicamente educativos que potencian el valor pedagógico que tienen las aventuras gráficas perse.

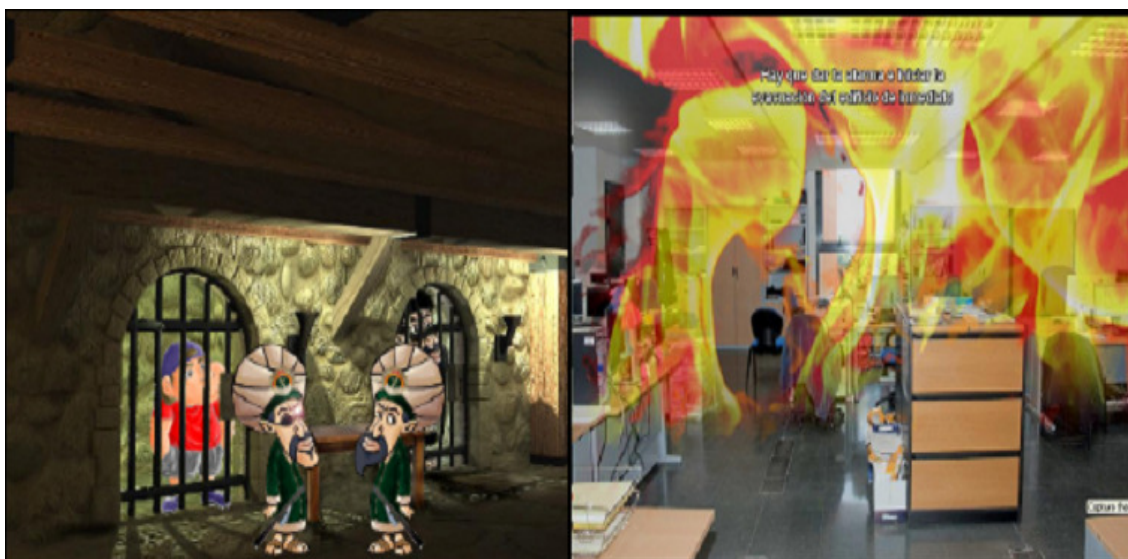


Figura 9: Dos ejemplos de juegos creados con la plataforma: aventuras en tercera y en primera persona.

Para dar una idea de cómo desarrollar un juego en <e-Adventure>, planteamos este proceso como el crear un guión de una obra de teatro. Para ello se deben definir los distintos elementos que lo conforman, como las escenas en las que tendrá lugar la acción de la aventura, el protagonista, los personajes que intervendrán, las conversaciones que podrán mantener y los objetos que aparecerán en las escenas. A partir de esta estructura se prepara el hilo narrativo. El editor brinda un mecanismo de control del estado de juego (variables y flags) que permiten guiar el desarrollo de la historia. Con los estados de juego se definen situaciones en las que el jugador puede cambiar de escena, puede ir manteniendo distintas conversaciones, recoger los objetos, etc. En definitiva estos estados rigen las distintas formas de interacción con los elementos del juego. Y en función de la interacción con dichos elementos se pueden ir planteando un conjunto de retos o puzzles que el alumno debe ir superando para avanzar en la historia.

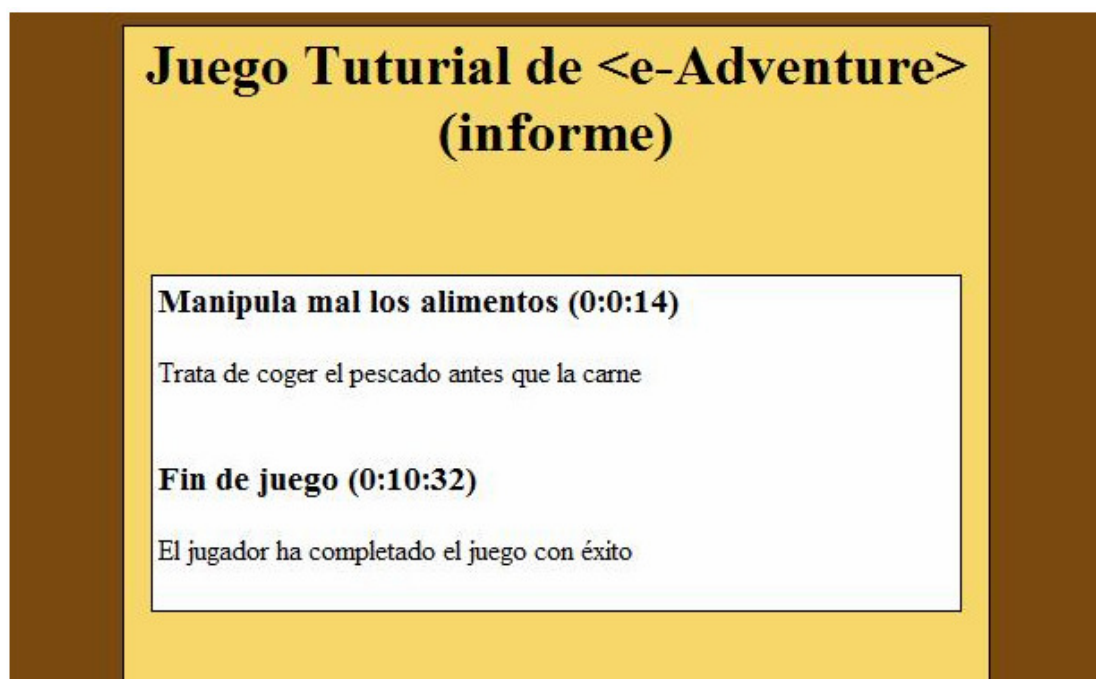
Existen dos tipos de juegos. Los juegos en primera persona, en los que no aparece el protagonista y están pensados para crear simulaciones basadas en juego, debido a que se puede simular un entorno real creando escenas a partir de fotografías. Estos juegos se plantean como una mejor alternativa a la enseñanza de procedimientos. El otro tipo son los juegos en tercera persona, en los que el jugador se identifica con un avatar. Son más adecuados para enseñar conceptos debido a que se pueden plantear mundos fantásticos en los que el jugador aprende a través del avance de la historia planteada.

Para añadir valor pedagógico a los juegos, <e-Adventure> presenta distintas formas de mostrar información al alumno. Además de las conversaciones, existen un tipo especial de escenas en las que se pueden mostrar transparencias o vídeos. Otra forma de mostrar información es mediante libros, permitiendo al desarrollador mostrar grandes cantidades de texto e imágenes como si de un libro real se tratase. Puede introducirse texto plano o con formato HTML. Estas formas de mostrar información tienen la ventaja añadida de permitir reutilizar contenidos que el profesor ya tuviese creados.

#### Características específicamente educativas

<e-Adventure> permite la evaluación y adaptación de los juegos creados con la plataforma (Moreno-Ger et al., 2007b; J. Torrente et al 2009). Mediante la monitorización del estado del juego de forma automática, se permite evaluación de la interacción del alumno por medio de un conjunto de reglas de evaluación definidas por el creador de la aventura. La interacción del alumno con el juego conlleva modificaciones de su estado. Las reglas definen estados de juego concretos para los cuales hay asociados una acción, la cual puede consistir en enviar determinada información al EVE o en rellenar un informe con datos al respecto de su actividad. El informe se puede remitir por correo al profesor, adjuntar al perfil del estudiante en el EVE y mostrar al final de la ejecución del juego al alumno para que le sirva como una posible autoevaluación.

Teniendo en cuenta las diferencias que existen entre los alumnos en relación a sus preferencias y capacidades de aprendizaje, <e-Adventure> incluye un mecanismo para modificar el desarrollo del juego. Permite adaptar el contenido a las particularidades concretas de cada alumno (ajustar la dificultad, amoldarlos a unas características socioculturales concretas y/o preferencias de aprendizaje, etc.) para maximizar la efectividad del juego para el mayor número de jugadores. Esto supone una ventaja en cuanto que permite obtener distintas versiones de un juego para usarlo en distintos contextos educativos sin modificar su estructura.



**Figura 10: Ejemplo de informe de evaluación generado por <e-Adventure>**

#### Integración de los juegos de <e-Adventure> en EVE

Otra característica interesante que favorece la utilización de videojuegos en la enseñanza es que se permite la exportación de los juegos como Objetos de Aprendizaje para que puedan ejecutarse y distribuirse en la mayoría de los EVE.

Como hemos ido analizando a lo largo de este documento, existen diferentes estándares para el empaquetamiento de los OAs. <e-Adventure> tiene distintos modos de exportación que permiten empaquetar los juegos en distintos formatos. Es objeto de estudio de este trabajo añadir nuevos estándares a las exportaciones de la plataforma.

## **4.2 Implementación de la arquitectura en la plataforma <e-Adventure>**

Los motivos por los que hemos elegido la plataforma <e-Adventure> para poner en práctica la arquitectura presentada son que posibilita tareas de adaptación del juego y evaluación de la interacción alumno, y que permite integrar los videojuegos creados en distintos EVE. Incluyendo la arquitectura en la exportación de los juegos creados para que sean integrados en un EVE podemos probar las funcionalidades de la SCA y la SCE en términos de adaptación y evaluación y la CC en términos de comunicación con el EVE. Analicemos en que ha consistido el proceso de implementación de la arquitectura en el ámbito de la plataforma <e-Adventure>.

Los términos abstractos del Modelo de Datos Abstracto de Adaptación y Evaluación no ha sido implementado en la plataforma debido a que actualmente se están estudiando las posibilidades que brinda y como materializarlas. Supone un gran desafío conseguir definir un conjunto de términos abstractos lo suficientemente amplio y flexible como para caracterizar la gran mayoría de las situaciones que se pueden presentar en términos de adaptación y evaluación. Debido a que su principal ventaja es facilitar las tareas de adaptación y evaluación a profesores sin conocimientos técnicos específicos al respecto, la

arquitectura puede funcionar. De todas formas se ha utilizado el modelo abstracto que brinda la plataforma que permite definir al creador de contenidos situaciones de juego concretas. Cabe destacar que si el creador del perfil es la misma persona que desarrollador del juego, o tiene conocimientos de la plataforma, no le supondrá ninguna dificultad añadida conocer el flujo del juego, ya que este se define en el editor de <e-Adventure>.

Antes de entrar en detalles en la implementación de cada una de las capas, creemos conveniente dar una idea de cómo afectan a la arquitectura las acciones realizadas desde que se crea el juego hasta su ejecución en un EVE. La arquitectura se configura mediante los perfiles de adaptación y evaluación y mediante la elección del tipo de exportación realizada en el editor de la plataforma. A continuación se exporta el juego eligiendo un estándar o EVE. El resultado de la exportación es un paquete que contiene los recursos necesarios para que pueda ser desplegado en el EVE: juego <e-Adventure>, recursos para la comunicación (HTML, JavaScript, etc.), metadatos y ficheros dependientes del estándar o EVE elegido. Una vez se ha desplegado el juego en el EVE, el juego comenzará a ejecutarse, entrando en funcionamiento la arquitectura.

#### 4.2.1 Implementación de la Capa de Adaptación y Evaluación

La plataforma <e-Adventure> lleva incluido un mecanismo de monitorización automático de la actividad del alumno con propósitos de evaluación y un mecanismo para realizar cambios en el y así adaptar la aventura. Por tanto, ya tiene implementadas las Subcapa de Adaptación y la Subcapa de Evaluación que, como ya hemos comentado, dependen del motor de juego específico en el que se va a integrar la arquitectura. Una vez creada la aventura y sobre todo, el flujo de juego, el autor puede añadir Perfiles de Adaptación y Perfiles de Evaluación en donde definir conjuntos de reglas de adaptación y evaluación con los propósitos anteriormente explicados. Estos perfiles han sido modificados para que tengan en cuenta las particularidades de los estándares de manera que se ajusten a la especificación de la arquitectura, amoldándose al concepto de Perfil de Adaptación y Perfil de Evaluación explicados en la sección 4.3. La única diferencia reside en que en la implementación de la arquitectura en <e-Adventure> no se materializa el MDAAE, por lo que no podemos expresar en términos abstractos ni las condiciones y ni las acciones de las reglas. Por tanto éstas estarán definidas en términos concretos del juego y del estado del EVE.

Para reflejar estados concretos de juego, <e-Adventure> permite definir un conjunto de variables y flags. La interacción del jugador cambiará estos valores de la manera que lo haya definido el autor del juego. Mediante su uso se pueden identificar situaciones concretas de juego. Además sirven para controlar el flujo, condicionando las interacciones con los objetos, con los personajes, las salidas a otras escenas, etc. Los flags son de tipo booleano y las variables son de tipo natural. Las condiciones son expresiones en algebra de bool en las que se evalúa distintos valores de flags y/o variables.

Analicemos en profundidad ambos perfiles y las modificaciones realizadas.

##### Perfil de Adaptación

En el Perfil de Adaptación encontramos los siguientes elementos:

- **Tipo de Perfil:** indica el tipo de estándar/EVE con el que se comunicará el juego exportado. En la actualidad hay tres tipos: Normal, SCORM 2004, SCORM v1.2.

Los tipos SCORM 2004 y SCORM v1.2 se corresponden con los módulos de estándar implementados en la CC en <e-Adventure>. De esta manera permite solicitar información almacenada en el modelo de datos del estándar elegido en el EVE para realizar tareas de adaptación. El tipo Normal sirve para aquellos estándares que no poseen modelo de datos, de manera que al elegir este tipo permite definirlo en las propias reglas, como Learning Design o LAMS.

- **Escena Inicial:** nos permite seleccionar un estado inicial del juego de manera incondicional. Esto facilita que con una simple modificación en esta parte del perfil el autor o cualquier profesor que abra con el editor un juego <e-Adventure> pueda proporcionar distintas versiones del mismo. Permite seleccionar una escena inicial y modificar el estado de juego.
- **Lista de Reglas de Adaptación:** aquí aparece un listado de las reglas de adaptación existentes, permitiendo crear, eliminar, cambiar el nombre y duplicarlas. Además se permite seleccionar la escena inicial en la que comenzará el juego en caso de que se cumplan las propiedades de la regla de adaptación.

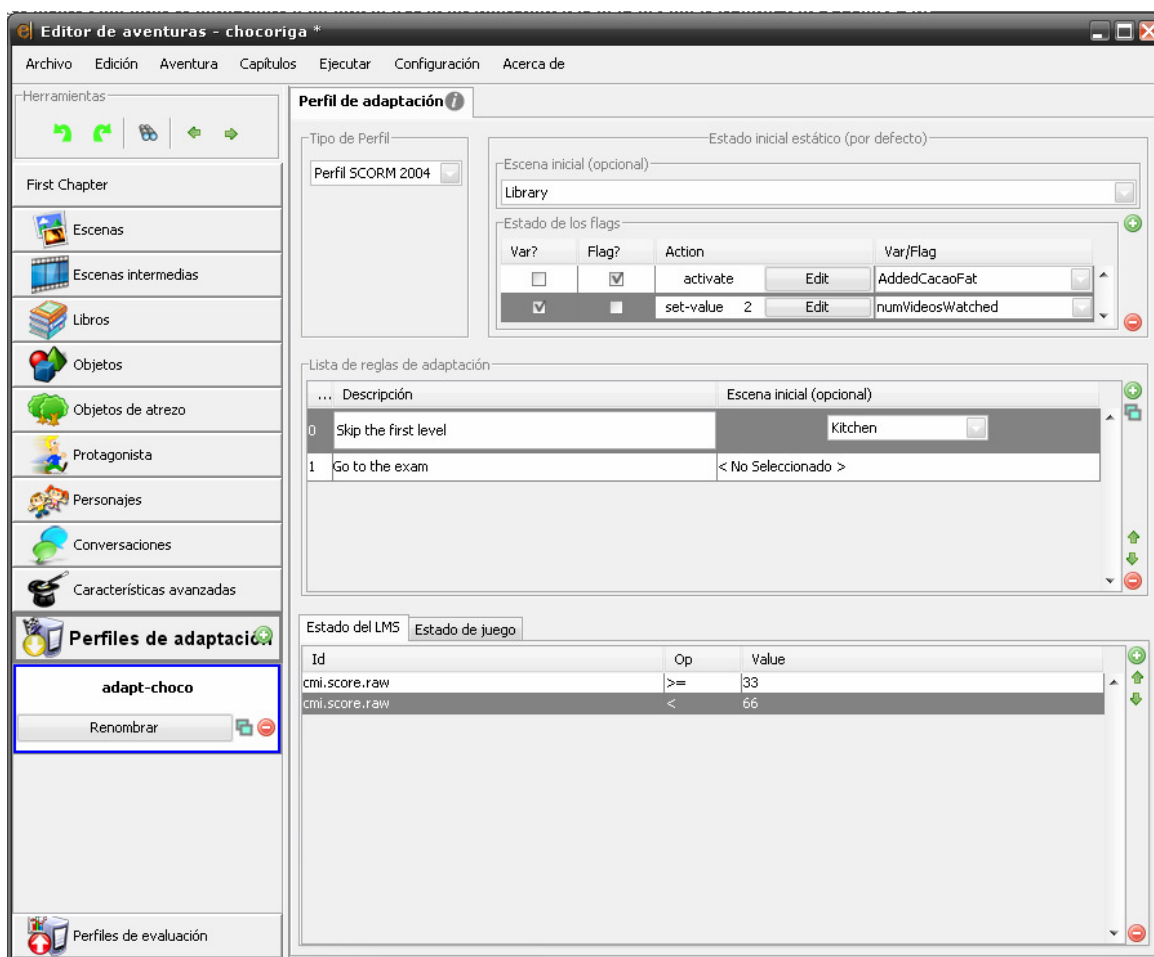


Figura 11: Ejemplo de configuración del perfil de adaptación.

- **Estado del EVE:** en esta pestaña se permite decidir que estados concretos del EVE condicionan que se cumpla la regla. En función del perfil seleccionado aparecerán las partes del modelo de datos a las que se puede acceder (en los perfiles SCORM), o permite definirlos desde cero (en el perfil Normal). Esta funcionalidad ha sido complementada, puesto que antes de implementar la arquitectura solo se daba soporte al perfil Normal. Para ello se ha incluido las partes del modelo de datos SCORM 2004 y v1.2 que son de lectura o de lectura/escritura. Además,

cuando se rellenan estos pares atributo/valor para un perfil de tipo Normal, a su vez se adjuntan en la parte que recoge el modelo de datos en el Perfil de Ajustes de Comunicación. Esta funcionalidad también ha sido añadida.

- **Estado del Juego:** en esta pestaña se permite decidir los cambios concretos que se producirán en el estado de juego una vez se cumplan las condiciones de la regla definidas en el “Estado del EVE”. Por tanto, las acciones asociadas a una regla quedan definidas por los datos rellenos en esta pestaña, y la escena inicial seleccionada en la “Lista de Reglas de Adaptación”.

### Perfil de Evaluación

En el Perfil de Evaluación encontramos los siguientes elementos:

- **Tipo de Perfil:** indica el tipo de estándar/EVE con el que se comunicará el juego exportado. Tiene la misma funcionalidad que el “Tipo de Perfil” en los perfiles de adaptación.
- **Configuración del Feedback al Alumno:** los perfiles de evaluación permiten adjuntar un informe de evaluación del alumno. En esta parte del Perfil de Evaluación se selecciona si el perfil será mostrado al alumno a modo de autoevaluación al terminar la aventura, y si será enviado al profesor, permitiendo rellenar el e-mail y la configuración SMTP del mismo.
- **Lista de Reglas de Evaluación:** aquí aparece un listado de las reglas de evaluación existentes, permitiendo crear, eliminar, cambiar el nombre y duplicarlas. Tienen un campo de “Importancia” que permite clasificar las reglas para mostrar en el informe solo aquellas que se soliciten. Existen dos tipos de reglas de evaluación. Las *reglas normales* se ejecutarán cuando se cumplan las condiciones asociadas. En el campo “Condiciones” se define que estado concreto debe darse en el juego para que se pueda ejecutar cada regla normal. Las *reglas temporizadas* disparan un temporizador cuando se cumplen unas condiciones iniciales y se detiene cuando se cumplen unas condiciones finales. Estas condiciones reflejan un estado concreto del juego. Para este tipo de reglas se pueden definir un conjunto de efectos. Cada uno de ellos lleva asociado un rango de tiempo en segundos. Cuando se detenga el temporizador se evalúan todos los efectos, y para aquellos que se cumpla el rango de tiempo definido, se ejecutará la acción asociada. Para ambos tipos de reglas, la acción asociada al cumplimiento de sus condiciones esta definida por los campos “Concepto”, “Texto” y “Modificación de las Propiedades”.
- **Concepto:** es la descripción corta del concepto que pretende expresar la regla de evaluación, y que será incluido como título de la regla en el informe de evaluación cuando dicha regla se cumpla.
- **Texto:** es una descripción mas extensa del propósito que evalúa la regla, y será incluido como contenido de la regla a continuación del título en el informe de evaluación cuando se cumpla la regla en cuestión.
- **Modificación de las Propiedades:** en esta parte del perfil permite seleccionar que datos se van a mandar al EVE cuando se cumplen las condiciones de la regla. En función del perfil seleccionado permite elegir la parte del modelo de datos a las que se puede acceder (en los perfiles SCORM), o permite definirlos desde cero (en el perfil Normal). De igual manera que en el “Estado del EVE”, esta funcionalidad ha sido mejorada con el soporte al modelo de datos SCORM 2004 y v1.2, incluyendo las partes del modelo que son de escritura o de lectura/escritura. Como también se

completa el modelo de datos del Perfil de Ajustes de Comunicación con los pares atributo/valor introducidos aquí cuando se trata de un perfil Normal.

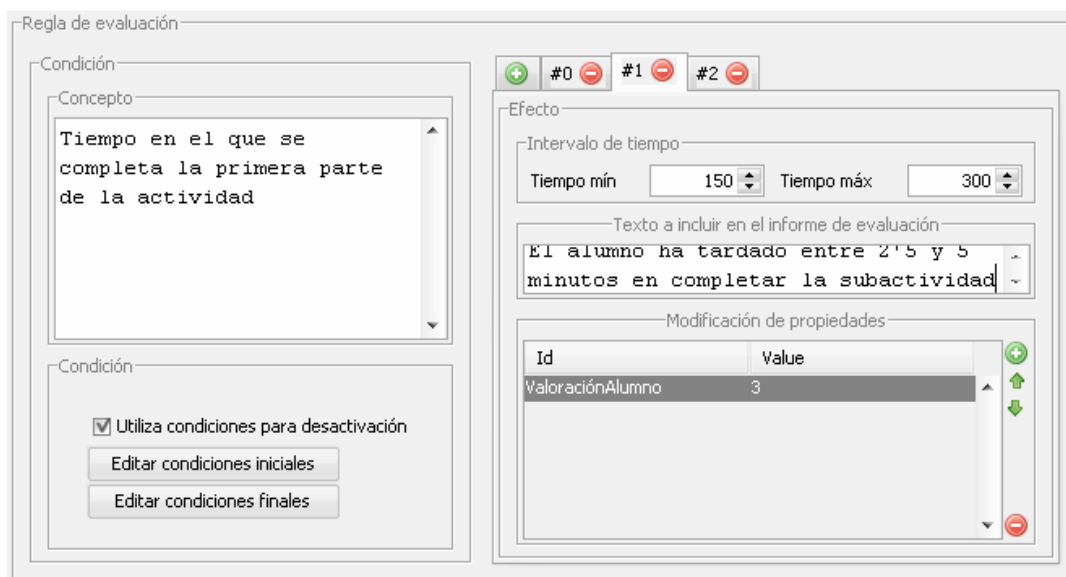


Figura 12: Ejemplo de la edición de las reglas temporizadas.

## 4.2.2 Implementación de la Capa de Comunicación

Para incluir en la plataforma la CC hemos realizado las siguientes tareas: implementar un módulo para cada estándar, dar soporte a los metadatos necesarios para los nuevos estándares incluidos y añadir nuevos tipos de exportaciones de los juegos <e-Adventure>. A continuación describimos con más detalle en lo que ha consistido cada una de estas tareas.

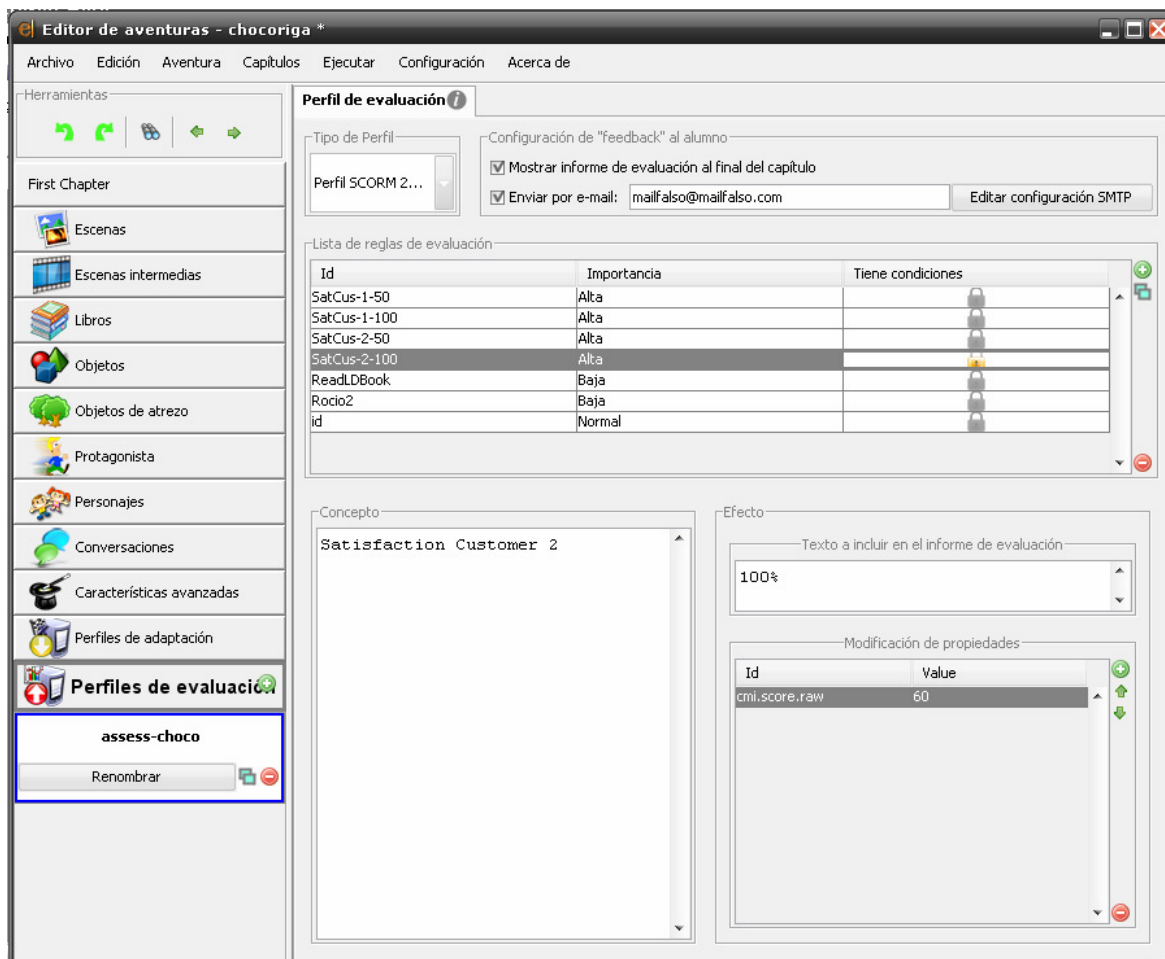
### Exportaciones como Objeto de Aprendizaje

Existen diferentes tipos de exportaciones como OA en referencia a distintos tipos de estándares y EVE. Se hacen exportaciones específicas para EVE puesto que pueden no adherirse a ningún estándar en concreto, o hacer pequeñas modificaciones sobre los mismos. En la plataforma ya se permitían exportaciones del tipo IMS-CP y WebCT-4.

El primer tipo de exportación que hemos añadido es el de SCORM v1.2. Para ello se ha utilizado como guía el libro técnico “The SCORM CAM” para la versión 1.2. El contenido en términos de organización de contenido SCORM tiene un Organización llamada “e-Adventure course” la cual está formada por un Ítem que tiene el título que se le haya dado al juego, y el recurso que tiene asignado es el HTML que contiene el applet que ejecutará el juego.

Otro tipo añadido se corresponde con SCORM 2004, utilizando mismo libro técnico pero para la versión 2004. En cuanto a nuestra implementación, las diferencias a la hora de generar las exportaciones de este tipo con las exportaciones SCORM v1.2 son: diferencias en el etiquetado del manifiesto y de los metadatos; cambios en los ficheros de esquema xml (.xsd); y diferencias en la nomenclatura de las funciones de la API. Mantiene la misma organización del contenido que SCORM v1.2.

Por último, en la actualidad nos encontramos finalizando la exportación de tipo LAMS, para permitir la exportación de juegos para que puedan ser incluidos como una actividad <e-Adventure> dentro de las secuencias de actividades. Para ello los paquetes se exportan como IMS-CP, con un conjunto de recursos extra que permiten la comunicación con LAMS, como se explica en la parte de módulos.



**Figura 13: Ejemplo de configuración del perfil de evaluación.**

Desde <e-Adventure> no existe una exportación per se para Learning Design. Como ya vimos en la sección 2, LD define un lenguaje para creación de metodologías de aprendizaje reutilizables, pero no define ni un protocolo de comunicación ni un modelo de datos. Esto no significa que las actividades que forman un Objeto LD no puedan comunicarse, simplemente LD no define el mecanismo de comunicación. LD se compone de un conjunto de Unidades de Aprendizaje (Units of Learning, UoL), las cuales están formadas por un conjunto de actividades empaquetadas como IMS-CP. Entonces sí se quisiera añadir un juego <e-Adventure> como una actividad de una UoL, este debería exportarse como IMS-CP y añadirse al Objeto LD en un editor de Objetos LD.

IMS-CP se centra solo en el empaquetamiento y etiquetado de los contenidos, por lo que no hay ningún mecanismo de comunicación. Es más, cuando se importa en un EVE un Objeto de Aprendizaje como IMS-CP, el EVE no espera que este se vaya a comunicar con él, por lo que no habilita ese servicio para este tipo de contenido. Esto motivó a añadir un protocolo de comunicación en las exportaciones IMS-CP en la plataforma, de manera que si el EVE permite la comunicación con un juego exportado como IMS-CP (situación que

se puede dar en LD), el juego pueda comunicarse. En caso de que no encuentre comunicación con el EVE, no habrá más inconveniente que no existirá comunicación juego-EVE. Este hecho se da para todos los tipos de exportaciones, aunque estén diseñadas para comunicarse: si el EVE no permite la comunicación, está no se realiza sin ningún inconveniente más que no podrán realizarse las tareas de evaluación y las de adaptación que requieran comunicación.

Una vez elegimos un tipo de exportación, se rellenarán los campos del Perfil de Ajustes de Evaluación. La siguiente tabla muestra cómo se rellena el Perfil de Ajustes de la Comunicación para cada tipo de exportación.

**Tabla 4: Relación entre los módulos creados para la capa de comunicación. Aparecen los valores con los que serán rellenado el Perfil de Configuración de la Comunicación.**

Tipo de Estándar/EVE	Tiene MD	Tiene PC	MD	PC
SCORM v1.2	Si	Si	--	--
SCORM 2004	Si	Si	--	--
LAMS	No	No	Recogido de los perfiles de adaptación y evaluación.	REST
IMS CP	No	No	Recogido de los perfiles de adaptación y evaluación.	AJAX

Antes de exportar deberemos asociar al proyecto de creación del juego los perfiles de adaptación y evaluación, para que se recoja la información que va a rellenar los Perfiles de Adaptación y Evaluación de la arquitectura. En un mismo proyecto se pueden crear distintos perfiles de distintos tipos. Hemos modificado el editor para que no permita exportar ningún juego a menos que los dos perfiles asociados sean de igual tipo y coincidan con el tipo de exportación. De esta manera evitamos posibles problemas de incompatibilidades en los Perfiles de Ajustes de Comunicación.

## Módulos

La CC consta de un conjunto de módulos con las particularidades de cada estándar. Cada módulo implementa los protocolos de comunicación que el estándar utiliza. En caso de que el estándar tenga definido un modelo de datos, también debe incluirse en la especificación del módulo.

Para implementar el módulo de LD en la plataforma se decidió implementar la API de la CC con tecnologías AJAX<sup>15</sup>. El modelo de datos quedará definido en los perfiles de adaptación y evaluación, como se vio en el apartado anterior. Versiones anteriores de <e-Adventure> daban soporte a este tipo de comunicación. Por ello simplemente hemos tenido que introducir en el módulo de LD el código que se utilizó con este fin. Cabe destacar que debe añadirse un plug-in al EVE de destino para que sea capaz de interpretar las llamadas que realiza la implementación de la API para LD (Torrente et al 2009).

<sup>15</sup> *Asynchronous JavaScript And XML*,

Otros de los módulos creados son los que dan soporte a SCORM 2004 y SCORM v1.2. Aunque ambos utilizan el mismo protocolo de comunicación (explicado en la sección 2.2.3) se ha creado un módulo para cada uno de ellos, ya que no tienen ni la misma nomenclatura para la API, ni las mismas funciones de error, ni el mismo modelo de datos. El editor consulta el modelo de datos cuando gestiona los perfiles de adaptación y evaluación para permitir al autor a decidir a que parte del mismo afectarán las acciones asociadas a las reglas que ahí se definen.

El último de los módulos añadidos es el que da soporte a LAMS. LAMS permite crear secuencias de actividades siguiendo las especificaciones de LD. Por ello, si una actividad desea comunicarse con este EVE, deben definirse un protocolo de comunicación y modelo de datos. Solventamos estas dificultades mediante la colaboración de ambos equipos de desarrollo. El protocolo de comunicación implementado en el módulo LAMS es REST<sup>16</sup>, que permite gestionar variables vía HTTP. De esta manera, se asigna una variable a una URL y se permite su acceso mediante las peticiones HTTP “GET” (recibir) y “POST” (enviar).

En cuanto al modelo de datos, será formado por las variables definidas en los perfiles de adaptación y evaluación, de manera que los juegos pueden utilizar variables definidas en la secuencia para adaptación y crear variables para evaluación de la actividad y adaptación de las siguientes actividades en la secuencia. Estas variables pueden ser utilizadas para realizar tareas del control de la secuencia (“gates”, “branching”). Al modelo de datos se añaden un conjunto de variables genéricas, todas de tipo cadena de caracteres, definidas por LAMS (ver tabla) que dan una información del perfil del estudiante en el EVE y de la actividad.

**Tabla 5: Variables por defecto que proporciona LAMS.**

Nombre	Descripción
firstname	Nombre del alumno.
lastname	Apellidos del alumno.
Group	Grupo al que pertenece el alumno.
userId	Identificador del usuario en el EVE.
lessonName	Nombre de la lección.
activityTitle	Título de la actividad.

Además, hemos definido un conjunto de variables predefinidas para los juegos <e-Adventure> y el informe de la actividad para caracterizar determinadas situaciones de juego con relevancia pedagógica (ver tabla). De esta manera se pretende facilitar la creación y el control de juegos y secuencias LAMS. Un posible uso es el de conocer si el alumno acabó o no el desafío del juego, o asignar valoraciones diferentes para distintos estados de finalización del juego. Además también se permite comprobar el tiempo total de juego, y el tiempo que el alumno ha dedicado realmente a jugar, es decir, no cuenta el tiempo que el juego permaneció en pausa o sin interacción. De esta manera nos da una idea del tiempo que el jugador estuvo interactuando con el juego.

**Tabla 6: Variables genéricas en <e-Adventure> para las secuencias LAMS.**

Nombre	Descripción
game-completed	Indica si se ha completado el juego o no.
Score	Valoración de la actividad del alumno.
report	Informe de evaluación <e-Adventure>.

<sup>16</sup> Representational State Transfer

totaltime	Tiempo de juego total (contando también cuando está parado y sin interacción del alumno).
Ingame-time	Tiempo real invertido en el juego.

Cuando el juego es exportado en cualquiera de los tipos comentados, antes de comenzar a ejecutar el juego se comunica al módulo responsable que realice las tareas pertinentes de inicialización de la comunicación. El módulo de SCORM v1.2 buscará la instancia de la API que el EVE deja en la estructura DOM de la página web que contiene el juego por medio de JavaScript y llamar a la función “LMSInitialize(“”)”. El módulo de SCORM 2004 realiza la misma tarea pero con las particularidades de su especificación, que básicamente se resumen en que la instancia de la API tiene otro nombre y utiliza otros nombres para sus funciones. Para inicializar utiliza la función “Initialize()” El módulo de Learning Design introduce la instancia de la API que utiliza para comunicarse en el DOM de la página vía AJAX. Volvemos a insistir en que hace falta añadir un plug-in al EVE para que interprete la información que se le envía. En LAMS las tareas de inicialización consisten en que LAMS proporcione las variables vistas en la tabla 5 y los valores del modelo de datos definido en el perfil de adaptación.

Cuando el juego se finaliza, el módulo de SCORM v1.2 llama a “LMSFinalize(“”)”, el de SCORM 2004 a “Terminate(“”)”, el de LD a una función en JavaScript para cortar la comunicación y en LAMS se devuelven las variables internas de <e-Adventure> comentadas en la tabla 6.

Figura 14: Vista del editor de metadatos para el estándar IEEE LOM

## Metadatos

Todas las exportaciones existentes en la plataforma llevan un conjunto de metadatos asociados para describir los juegos <e-Adventure>. Los metadatos ayudan a la clasificación y almacenamiento en repositorios de Objetos de Aprendizaje y EVE y dan una idea del

propósito para el que fue creado el OA con el propósito de que sean reutilizados. En <e-Adventure> se soportan dos tipos de edición de metadatos:

- **Metadatos IMS Learning Resource Meta-data v1.2:** Este es el tipo de metadatos que utiliza la especificación SCORM v1.2. Tiene ligeras diferencias con el estándar IEEE/LTSC LOM, por lo que debimos añadir el editor de este tipo de metadatos para poder añadir metadatos a las exportaciones SCORM v1.2.
- **Metadatos IEEE/LTSC LOM:** Este tipo de metadatos lo utilizan la especificación SCORM 2004 e IMS CP. Ya estaba implementado este editor de metadatos, debido a que la exportación de juegos como IMS CP es anterior a la implementación de la arquitectura.

Para ambos editores no se implementan todos los campos, sino que se permiten editar los que el estándar obliga a que existan más una selección de los que consideramos más interesantes.

### 4.3 Conclusiones del Capítulo

<e-Adventure> es la plataforma de juegos escogida para comprobar en un tipo de juegos concretos las ideas teóricas planteadas para la arquitectura. Se realizó la implementación de las capas que conforman la arquitectura de manera que se incluyan en los juegos creados con la plataforma. Se añadió al editor de juegos los elementos necesarios para configurar la arquitectura, de igual manera que se crearon tipos de exportación de juegos específicos para los estándares a los que se da soporte.

Una vez que se da soporte a la arquitectura, parece razonable crear juegos y configurarlos para que se puedan comunicar con distintos estándares. Por ello, se presenta un caso de estudio en el que se pone a prueba la integración de un juego modificado en un EVE para comprobar que realiza correctamente las tareas de adaptación y evaluación.

## 5. Caso de Estudio

Para comprobar que la implementación de la arquitectura se ajusta al funcionamiento explicado en la propuesta se plantea un caso de estudio. Utilizamos un juego <e-Adventure> ya creado y completamos la configuración de la arquitectura (Perfiles de Evaluación, Adaptación y de Ajustes de la Comunicación). En este caso de estudio se plantean diversas situaciones de evaluación y adaptación. Al iniciar el juego el estado del EVE no refleja ninguna actividad del alumno. De esta manera proponemos diversos estados de juego para diversas evaluaciones de la interacción del alumno en el juego y comprobamos si realmente se llegan a las situaciones deseadas en diversas ejecuciones.

El caso de estudio se centra en los siguientes aspectos:

- Comprobar que se realiza correctamente la *configuración* de la arquitectura.
- Comprobar que se realiza correctamente la *comunicación* entre todos los elementos implementados que entran en juego en la arquitectura, es decir, entre EVE y la CC, entre la CC y la CAE. La comunicación entre el EVE y la CC se realiza utilizando distintos estándares.

Por tanto nos centramos en la comunicación EVE-juego. En este caso de estudio no abordamos probar distintos tipos de adaptación (en función de razones socio-culturales, de accesibilidad, de preferencias de aprendizaje, etc.) porque queda fuera del alcance de este trabajo. La comunicación entre la CAE y el juego <e-Adventure> ya estaba implementada y su funcionamiento está comprobado (Torrente et al 2009), solamente se ha acoplado a la estructura de la arquitectura. Como ya hemos comentado, la SCA y la SCE dependen de determinados factores específicos del motor de juego.

### 5.1 Presentación del caso de estudio

El juego que utilizamos se llama “*Paniel and the Chocolate-based Sauce Adventure*” en el que el protagonista se apunta a una academia de cocina para aprender todo sobre el chocolate. Durante el desarrollo del juego, el alumno aprenderá distintos aspectos sobre el chocolate (historia, como hacerlo, como usarlo en distintas recetas, etc.) desde un punto de vista práctico, es decir, mientras supera los distintos retos que se le proponen. En este juego están claramente diferenciados tres niveles. En el primero, el alumno aprenderá como hacer chocolate. En el segundo, se enseña a combinar chocolate con diversos productos para crear salsas. Finalmente, el tercer nivel explica como combinar las salsas creadas con distintos platos.

Los estándares elegidos para el caso de estudio son SCORM v1.2 y SCORM 2004. El EVE elegido es Ilias ya que soporta las dos versiones de SCORM y tiene el certificado de ADL de compatibilidad SCORM. Además se probará integrar en un Moodle la versión del juego para SCORM v1.2, ya que sólo soporta esta.

### 5.2 Detalle del caso de estudio

Para el caso de estudio hemos identificado distintas situaciones de juego de relevancia para dar una evaluación del alumno, las cuales se identifican en el ejemplo por una “E” seguida de un número. Esta evaluación quedará registrada en el modelo de datos de cada una de las

versiones de SCORM. Hemos definido un conjunto de acciones que se deben realizar en el juego en función de la actividad registrada en el EVE para sucesivas ejecuciones del juego, de manera que si se han alcanzado determinados objetivos se modifique el transcurso del juego. Estas se identifican mediante una “A” seguida de un número.

Para completar el primer nivel el jugador debe crear tres tipos distintos de chocolate y entregárselos a su tutor en el juego. Vamos a observar el funcionamiento de evaluación, como lo comunica al EVE, y como se adapta el juego en la siguiente ejecución en función de la evaluación recogida. Hemos considerado tres casos, mostrando distintos formas de adaptar el juego. En el primero, el resultado de la adaptación del juego no modifica la escena inicial pero lleva a dos estados de juego distintos. La segunda, llevará a distintas escenas. La tercera es el resultado de unir los dos casos anteriores. Estas son los distintos casos que hemos tenido en cuenta:

#### Caso 1:

Identificamos dos situaciones en la que se debe realizar evaluación:

- **E1:** Si ha entregado los tres tipos de chocolate cometiendo un error a la hora de mezclar los ingredientes. Además no leyó el libro de cómo hacer chocolate.
- **E2:** Si ha entregado los tres tipos de chocolate cometiendo dos o más errores y no se leyó el libro de chocolate.

Imaginemos que el jugador abandona el juego en este punto, o no realiza ninguna acción relevante más. En la próxima ejecución el juego se modificará para adecuarse a las destrezas que ya posee el jugador. Estas son las dos posibles situaciones:

- **A1:** Comprueba que se ha dado la situación **E1**. Comienza en la escena inicial. Para salir de ella deberá hablar con la recepcionista, la cual le explicará que debe aprender a hacer salsas de chocolate. Además aparecerá en su inventario un libro con una receta de una salsa que le servirá para completar el siguiente nivel.
- **A2:** Comprueba que se ha dado la situación **E2**. Empieza en la misma escena que en “**A1**”, pero el al hablar con la recepcionista indica al alumno que debe ir a la cocina a aprender a hacer chocolate. Además aparecerá en su inventario los ingredientes necesarios para su elaboración, evitando que tenga que volver a recogerlos.

#### Caso 2:

En este caso se evalúan dos estados de juego:

- **E3:** el jugador ha entregado los tres tipos de chocolate sin ningún error, pero tardó entre 2 y 5 minutos desde que leyó el libro de cómo hacer chocolate hasta que entregó el último tipo.
- **E4:** El jugador ha entregado los tres tipos de chocolate sin ningún error, ha consultado el libro de cómo hacer chocolate y ha tardado menos de 2 minutos en entregar todos los tipos. Además ha leído el libro de recetas de salsas de chocolate antes de intentar realizar una salsa, independientemente de que consiguiera o no que saliera bien.

Estos son los cambios que se realizarán en el juego en función del estado del modelo de datos:

- **A3:** Comprueba que se ha dado la situación **E3**. El juego empezará en la biblioteca, en donde iniciará una conversación con el tutor en el que le indica que se lea el libro de recetas de salsas de chocolate para poder avanzar en el juego.
- **A4:** Comprueba que se ha dado la situación **E4**. El jugador pasa al nivel 3 y no tendrá que leer el libro de recetas, acción obligada para continuar en la historia de ese nivel si se hubiera empezado el juego desde el principio. El juego comenzará en la cocina y no tendrá nada en el inventario.

### Caso 3

En este caso es el resultante de unir los dos anteriores.

## 5.3 Configuración de las capas

Como vamos a probar las dos versiones de SCORM implementadas en la arquitectura tendremos que generar un perfil de evaluación y uno de adaptación para cada una de ellas, asignándole el tipo en concreto que vamos a usar. De esta manera el editor rellena el campo “Tipo de Estándar/EVE” con SCORM v1.2 y SCORM 2004 respectivamente. El editor rellena automáticamente los campos “Tiene modelo de datos” y “Tiene protocolo” a cierto para ambas versiones. No hace falta rellenar ningún campo mas, por lo que el Perfil de ajustes de la Comunicación queda correctamente configurado. La configuración de la CC es la misma para los tres casos.

Vamos a analizar la configuración de la CAE para el Caso 3 ya que es este caso es la unión de los otros dos. Para configurar el Caso1 solo tendría que rellenarse el perfil de evaluación con E1 y E2, y el de adaptación con A1 y A2. El Caso 2 quedaría reflejado rellenando los perfiles de adaptación y evaluación con A3-A4, y E3-E4 respectivamente.

Para configurar la SCE rellenaremos el perfil de evaluación. Para ello crearemos una regla que refleje cada uno de las situaciones explicadas. La regla **E4** vamos a descomponerla en dos reglas ya que consideramos que representa dos situaciones relevantes de evaluación, una asociada a cada objetivo. Ambas situaciones no se dan al mismo tiempo, y puede que ocurra una y no ocurra la otra. La regla **A4** dependerá del estado de los dos objetivos para comprobar que se pueden hacer chequeos de varios campos del modelo de datos de SCORM.

Antes de ver su configuración veamos los elementos concretos que permiten identificar las situaciones de juego expresadas. En el desarrollo del juego se creó un flag (3chocoOK) para identificar cuando el jugador entregó correctamente los tres tipos de chocolate que le fueron solicitados. Esto permite avanzar de nivel en el juego dando por superado este “puzzle” en concreto. Además se definió una variable para contar el número de veces que el jugador comete un error a la hora de crear el chocolate, llamada numErr1. Otro flag que entra en juego en estas situaciones es “leyoLibroChoco” que se activa tras leer el libro de cómo crear por primera vez. Solo se permite leer este libro si no se han entregado los chocolates. Por último, el flag “leyoLibroRecetas” indica que se ha leído el libro de recetas por primera vez y el flag “inicioRecetas” indica que se intentó hacer alguna receta con chocolate.

Para poder controlar el tiempo que tarda el alumno desde que lee el libro hasta que entrega los tres tipos de chocolate se utiliza un temporizador en vez de una regla temporizada. Esta decisión viene motivada porque la regla E4 no solo depende del intervalo de tiempo que registra una regla temporizada, sino que entra en juego determinada actividad del alumno durante el transcurso de ese intervalo. Un temporizador permite definir al autor del juego unas condiciones iniciales que lo activarán y unas finales que lo desactivarán. Permite realizar cambios en el estado del juego debido a que se hayan cumplido sus condiciones de desactivación o por que haya transcurrido un determinado periodo de tiempo definido. Se puede configurar en bucle, de manera que no para de funcionar hasta que no se cumplan sus condiciones finales. Para el juego definimos un temporizador con los siguientes parámetros: como condición inicial que se lea el libro de cómo hacer chocolate, como condición final que se hayan entregado los tres tipos de chocolate, con 1 minuto de tiempo, y que cada vez que acabe ese minuto incremente una variable llamada “minHastaOKChoc” en una unidad.

**Tabla 7: Relación entre reglas de evaluación, el estado de flags que suponen su activación y los cambios que se enviarán al EVE.**

Regla de Eval.	Condición	Acciones para SCORM v1.2	Acciones para SCORM 2004
E1	3chocoOK && numErr1 == 1 && lleyoLibroChoco	LMSSetValue(“cmi.objectives.0.id”, treschoc); LMSSetValue(“cmi.objectives.0.score.raw”, “55”) LMSSetValue(“cmi.objectives.0.status”, “completed”)	SetValue(“cmi.objectives.0.id”, treschoc) SetValue(“cmi.objectives.0.score.raw”, “55”) SetValue(“cmi.objectives.0.completion_status”, “completed”)
E2	3chocoOK && numErr1 >= 2 && lleyoLibroChoco	LMSSetValue(“cmi.objectives.0.id”, treschoc); LMSSetValue(“cmi.objectives.0.score.raw”, “40”) LMSSetValue(“cmi.objectives.0.status”, “failed”)	SetValue(“cmi.objectives.0.id”, treschoc) SetValue(“cmi.objectives.0.score.raw”, “40”) SetValue(“cmi.objectives.0.completion_status”, “failed”)
E3	numErr == 0 && minHastaOKChoc > 2 && minHastaOKChoc <= 5	LMSSetValue(“cmi.objectives.0.id”, treschoc); LMSSetValue(“cmi.objectives.0.score.raw”, “75”) LMSSetValue(“cmi.objectives.0.status”, “completed”)	SetValue(“cmi.objectives.0.id”, treschoc) SetValue(“cmi.objectives.0.score.raw”, “75”) SetValue(“cmi.objectives.0.completion_status”, “completed”)
E4a	numErr == 0 && leyoLibroChoco && minHastaOKChoc <= 2	LMSSetValue(“cmi.objectives.0.id”, treschoc); LMSSetValue(“cmi.objectives.0.score.raw”, “85”) LMSSetValue(“cmi.objectives.0.status”, “completed”)	SetValue(“cmi.objectives.0.id”, treschoc) SetValue(“cmi.objectives.0.score.raw”, “85”) SetValue(“cmi.objectives.0.completion_status”, “completed”)
E4b	leyoLibroRecetas && inicioRecetas	LMSSetValue(“cmi.objectives.1.id”, recetas); LMSSetValue(“cmi.objectives.1.score.raw”, “15”) LMSSetValue(“cmi.objectives.1.status”, “incomplete”)	SetValue(“cmi.objectives.1.id”, recetas) SetValue(“cmi.objectives.1.score.raw”, “15”) SetValue(“cmi.objectives.1.completion_status”, “incomplete”)

Definimos dos objetivos para el modelo de datos de cada versión de SCORM para identificar dos metas a cumplir en el juego. El primero refleja que se entregan los 3 tipos de chocolate, dándole a cada situación una valoración e indicando si se completó o no el objetivo. El segundo refleja que se elaboró correctamente por lo menos una salsa. Cada uno de los objetivos del modelo de datos SCORM esta relacionado con los objetivos que se

desea que se cumplan en los dos primeros niveles del juego. Siguiendo las indicaciones dadas por SCORM, antes de rellenar ningún campo de un objetivo hay que crearlo almacenando el valor de su identificador. Por ejemplo, en SCORM 2004 el primer objetivo se creará mediante una llamada a SetValue(“cmi.objectives.0.id”,”treschoc”). SCORM permite crear el mismo objetivo dos o mas veces si y solo el identificador y el número de objetivo son idénticos. Además SCORM obliga a que se creen los objetivos en orden. Tal y como lo hemos definido en las acciones asociadas a las reglas de evaluación se cumple con esta obligación.

La configuración de la SCA la realizamos creando reglas que reflejen los estados de EVE que hemos decidido que supondrán modificaciones en el juego para adaptar la experiencia educativa. Cada una de las reglas de adaptación depende de estados que reflejaron las reglas de evaluación. La siguiente tabla muestra que cambios se realizarán en el estado de juego (variables, flags y escenas iniciales) para modificar el juego como se definió en la sección anterior.

**Tabla 8: Relación entre reglas de adaptación, la condición que activa la regla en función del estado del EVE, y los cambios concretos en el juego (modificación de flags).**

Regla de Adaptación	Condición	Escena Inicial	Modificación del Estado del juego
A1	cmi.objectives.0.score.raw == 55	Escena inicial del juego (Hall)	ActivaConv1
A2	cmi.objectives.0.score.raw == 40	Escena inicial del juego (Hall)	ActivaConv2
A3	cmi.objectives.0.score.raw == 75	Biblioteca	ActivaConv3
A4	cmi.objectives.0.score.raw == 85 && cmi.objectives.1.completion_status == “incomplete”	Cocina	noMostrarLibro && nivel3

Cuando se cumple la condición de A4, se modifican los dos flags mostrados en la tabla para adaptar el juego. No se necesitan más cambios ya que en el juego original estaban claramente delimitados los niveles, de manera que al activar el flag “nivel3” el juego pasa a ese estado. Además al activar el flag “noMostrarLibro” se evita que el jugador tenga que leer el libro para continuar en la historia, por lo que queda reflejada la situación descrita en **A4**.

Los flags que se modifican para las reglas **A1**, **A2** y **A3** permiten que se lancen unas conversaciones que se han añadido ad-hoc al juego para realizar los cambios en función de las situaciones de adaptación planteadas. En cada una de las distintas conversaciones se ejecutan un conjunto de flags que modifican la historia del juego para amoldarse a cada situación concreta.



## 6. Conclusiones y Trabajo Futuro

En este trabajo se ha abordado principalmente cómo se pueden integrar juegos adaptativos en entornos de enseñanza preexistentes. En el problema planteado se identifican dos frentes: la diversidad de tipos de juegos y de servidores educativos. Además es interesante que para aumentar el valor pedagógico de los juegos se facilite el acceso a los propios docentes al proceso de modelado y diseño del comportamiento adaptativo de los juegos y la estrategia de evaluación a seguir, participando activamente en la toma de decisiones de qué situaciones del juego deben ser evaluadas y de qué manera. La solución planteada es una arquitectura multicapa para paliar el problema de la variedad de juegos y estándares y proveer un mecanismo que facilite la participación de expertos en el diseño de la toma de decisiones de adaptación y evaluación. La idea de que sea “multicapa” se adoptó para afrontar de manera independiente los problemas asociados a afrontar los frentes comentados.

Para que la aplicación de las soluciones propuestas sea suficientemente general la arquitectura debe dar soporte a un conjunto significativo de estándares y especificaciones con el propósito de que la comunicación sea transparente al creador de videojuegos. La arquitectura presenta una capa encargada de comunicarse con los distintos tipos de EVE. Se realiza un estudio de los estándares en *e-Learning*, y más concretamente de SCORM, para poder dar solución al problema de la comunicación e implementar los distintos módulos encargados a tal fin. Este estudio ha servido también para conocer los modelos de datos que utilizan estos estándares y las particularidades de los distintos servidores educativos.

La arquitectura presenta una capa encargada de realizar evaluación y adaptación en el juego. Trabaja de forma conjunta con la capa de comunicación para enviar evaluaciones y para realizar peticiones del estado de la interacción del jugador. En función de los valores recibidos, modifica de una u otra manera el desarrollo del juego. La parte de esta capa encargada en monitorizar y materializar cambios en el juego está muy ligada al tipo de juego utilizado. La implementación de estas características en distintos motores de juego queda fuera del alcance de este trabajo.

Además se plantea desde un punto de vista teórico la estructura y relación con el resto de capas del modelo de datos abstractos para facilitar a los docentes la aplicación de determinadas metodologías educativas. La idea es definir de manera genérica un conjunto de términos que permitan identificar un conjunto de situaciones en el juego que sean de relevancia para ser evaluadas. Y de igual manera, definir un conjunto de términos que identifiquen determinados cambios en el desarrollo del juego.

En este trabajo se presenta la aplicación de las ideas planteadas en la arquitectura en la en una plataforma en concreto (<e-Adventure>) para comprobar sus funcionalidades. Además se añade la posibilidad de configurar sus características para exportar casos concretos de juegos con el objetivo de integrarlos en determinados sistemas educativos. Una vez realizado este proceso, se puso en prueba el funcionamiento de las distintas capas en un juego en concreto y en distintos EVE.

### 6.1 Resultados

Las principales aportaciones de este trabajo son:

1. Revisión y análisis crítico de los estándares de *e-learning* y su uso con propósito adaptativo. Revisión de EVE y comparativa entre algunos de los más utilizados hoy en día. En este estudio tiene un papel preponderante la propuesta SCORM debido tanto a su aceptación en el mercado como al hecho de que es la única que propone un modelo de ejecución, fija cual es el modelo de comunicación entre los contenidos y el EVE y plantea el uso de un modelo de datos concreto (e.g. CMI). Además, propone un modelo de aplicación del estándar IMS Sequencing Simple que permite al diseñador de las experiencias educativas condicionar el flujo de las actividades teniendo en cuenta las interacciones del alumno. Aunque este no es el mecanismo utilizado en la arquitectura para realizar adaptación, ha servido para conocer la propuesta realizada por una organización de la importancia de ADL.
2. Propuesta de una arquitectura genérica de integración de juegos educativos en entornos virtuales de enseñanza. Esta arquitectura tiene en cuenta entre otros estándares el modelo SCORM para realizar la comunicación y obtener soluciones generalizables a distintos juegos y EVE. Un modelo de plug-ins que permite contemplar otras especificaciones y protocolos actuales, evoluciones y nuevas versiones de los mismos e incluso nuevos modelos no previamente contemplados (e.g. LAMS). La propuesta contempla un módulo de adaptación para transformar la información recibida del EVE en cambios concretos en el juego y así amoldar la experiencia educativa en función de la actividad del alumno. También presenta un módulo de evaluación que monitoriza la interacción alumno-juego con el fin de enviar evaluaciones al servidor educativo. Además incluye un modelo de datos abstractos que define un conjunto de términos para conceptualizar las tareas de adaptación y evaluación y alejarlas de las dificultades técnicas que supone su implantación en juegos y servidores.
3. Aplicación de la arquitectura genérica a un caso concreto que es el entorno <e-Adventure>. Para llevarlo a cabo se han tenido que realizar modificaciones y ampliar funcionalidades de la plataforma. En primer lugar encontramos el desarrollo del módulo de comunicación. Incluye el acondicionamiento de la implementación de los estándares ya existentes en la plataforma (e.g. Learning Design) a esta nueva estructura y se ha ampliado con las especificaciones SCORM v1.2 y 2004 y con el plug-in para integrar juegos <e-Adventure> en LAMS. Otro aspecto abordado ha consistido en realizar diversas mejoras en los procesos de exportación de la plataforma. Los cambios realizados supusieron dar soporte al empaquetamiento de juegos para SCORM y LAMS y ampliar el editor de metadatos de la plataforma. Por último se realizaron diversas mejoras en el editor para permitir la configuración de la arquitectura. Entre ellas encontramos las modificaciones en la edición de los perfiles de evaluación y adaptación para reflejar los modelos de datos de las distintas versiones de SCORM.
4. Ejemplificación y prueba de la aproximación en un caso concreto de juego. Para ello utilizamos un juego ya creado con la plataforma <e-Adventure>. Una vez implementadas las particularidades de la arquitectura en la plataforma, se realizaron distintas modificaciones en el juego para comprobar las funcionalidades añadidas. Estos cambios consistieron principalmente en configurar distintas características de adaptación y evaluación y prepararlos para distintos estándares. Una vez concluido, se realizó la exportación del juego configurado para ser compatible con SCORM v1.2 y 2004. Ambas exportaciones se probaron en el servidor educativo Ilias, el cual tiene la certificación oficial de ADL para ambas versiones de SCORM. También se probó el juego en el EVE Moodle, pero sólo para la versión 1.2 de SCORM, ya que este servidor educativo sólo es parcialmente compatible con este perfil de aplicación. Los resultados fueron bastante positivos. Para todas las importaciones

realizadas se observaron los comportamientos esperados. Para las situaciones de evaluación añadidas se comprobó que quedaron reflejados en el modelo de datos CMI de SCORM que posee cada EVE los datos concretos asociados a los estados del juego reflejados. Además se comprobó que para cada situación concreta del modelo de datos CMI en el EVE se llegaron a los estados deseados de juego expresados en las situaciones de adaptación incluidas, consiguiendo modificar la estructura del mismo en función pasadas interacciones del jugador. Por tanto mediante la implementación de las ideas planteadas en la arquitectura para las particularidades concretas de un tipo de juegos se facilita la integración de este tipo de contenidos educativos en distintos EVE comunicándose con ellos para realizar tareas de adaptación y evaluación. Solamente deben modificarse los perfiles que configuran las capas que forman la arquitectura para conseguir integrarlos en distintos servidores educativos, alejando a los creadores de las dificultades técnicas que esto supone.

## 6.2 Trabajo Futuro

En un trabajo de investigación siempre quedan aspectos que hay que seguir desarrollando para lograr mejores soluciones. En este caso los siguientes pasos a dar son:

1. Modelo abstracto de adaptación. Aunque se ha proporcionado una solución funcional, ésta todavía requiere un gran conocimiento de la dinámica concreta y de los detalles del juego para poder realizar la adaptación al usuario. Sería deseable identificar un conjunto de patrones que permita simplificar esta tarea y expresarla en términos más abstractos (e.g. aumentar o disminuir el nivel de dificultad) sin que necesariamente se conozca de antemano en qué acciones concretas se reflejan los cambios en cada juego específico. Además, la identificación de las situaciones concretas en las que se debe adaptar el contenido de juego es todavía una línea de investigación abierta.
2. Modelo abstracto de evaluación. Actualmente aunque la información se puede utilizar con propósitos de evaluación sigue siendo necesario tener un gran conocimiento de la dinámica concreta de cada juego. Sería deseable encontrar una expresión más abstracta y genérica que permita evaluar distintos factores de rendimiento en la interacción del alumno. Actualmente esta fase se ha comenzado ya en colaboración con los desarrolladores del entorno LAMS en Macquarie University en Australia.
3. Generalizar los resultados actuales obtenidos para la plataforma <e-Adventure> a otros modelos y plataformas de juegos. Para ello es necesario implementar las particularidades de cada uno de ellos en la capa de adaptación y evaluación de manera que permita monitorizar la interacción del alumno y realizar modificaciones en el juego. La idea es que se puedan añadir distintos módulos específicos e independientes para solventar las características concretas de cada plataforma de juego, siguiendo la misma idea utilizada en la capa encargada de la comunicación.
4. Integración completa con Moodle una vez que dicho entorno sea cien por cien compatible SCORM. Actualmente solo refleja una parte del modelo de datos de este modelo de aplicación de SCORM v1.2, de manera que no se puede explotar el potencial que este presenta para llevar a cabo evaluación del alumno.
5. Diseño de estrategias genéricas de uso de los datos obtenidos en el juego para modificar el flujo educativo en el EVE. En la implementación del módulo LAMS explicado se pueden intercambiar un conjunto de variables predefinidas en la plataforma <e-Adventure> para las cuales se ha dado un significado concreto. Usando estas variables se puede condicionar el flujo de la experiencia educativa. De

esta manera los juegos podrían ser reutilizados por los educadores en distintas situaciones educativas sin tener que ser expertos conocedores del diseño y detalles del propio juego.

6. Creación de un framework de desarrollo para implementar las ideas planteadas en la arquitectura para facilitar su aplicación en otros contenidos educativos distintos a los juegos.

## 7. Referencias

- Adell, J. (1997). *Tendencias en la educación y en la sociedad de las tecnologías de la información*. EDUTEC Revista Electrónica de Tecnología Educativa (7), <http://www.uib.es/depart/gte/revelec7.html>
- ADL. (2006). *Advanced Distributed Learning Sharable Content Object Reference Model (ADL-SCORM)*. Retrieved October 7, 2008 from <http://www.adlnet.org/>.
- Aldrich, C. (2005). *Learning by Doing: A Comprehensive Guide to Simulations, Computer Games, and Pedagogy in e-Learning and Other Educational Experiences*. San Francisco, CA: Pfeiffer.
- Barker, J. R. (1999). *The discipline of group work*. London, England. Sage.
- Brusilovsky, P. (2001). *Adaptive Hypermedia*. *User Modeling and User-Adapted Interaction* (11), 87-110.
- Chen, J., T. Wang, W. Chang, y L. R. Chao (2009). *Developing an Interactive Video-Based Learning Environment*. *Journal of Software*, 4(2).
- Conlan, O., V. Wade, M. Gargan, C. Hockemeyer, y D. Albert (2002). *An architecture for integrating adaptive hypermedia services with open learning environments*. *Proceedings of ED-MEDIA'2002- World Conference on Educational Multimedia, Hypermedia and Telecommunications, Denver, CO (USA) AACE*, 344-350.
- de Antonio, A., J. Ramírez, y Z. Madrigal (2008). *Architecture for Integration of Simulations with SCORM in E-learning Environments*. In *Proceedings of The 8th IEEE International Conference on Advanced Learning Technologies - ICALT 08, Santander, Cantabria (Spain)*, 217-221
- de Freitas, S. y M. Oliver (2006). *How can exploratory learning with games and simulations within the curriculum be most effectively evaluated?* *Computers & Education* 46(3), 249-264.
- Dalziel, J (2003). *Implementing Learning Design: The Learning Activity Management System (LAMS)*. In: Crisp, G., Thiele, D., Scholten, I., Barker, S., Baron, J. (eds.): *20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education*. ASCILITE [www.ascilite.org.au](http://www.ascilite.org.au), Adelaide, Australia.
- Dillon, A. y R. Gabbard (1998). *Hypermedia as an educational technology: a review of the empirical literature on learner comprehension, control and style*. *Review of Educational Research*, 68(3), 322-349.
- Duart, J. y A. Sangrà (2000). *Aprender en la Virtualidad*. Barcelona, Spain. Gedisa, Edicions de la Universitat Oberta de Catalunya. 253
- Echazarreta, C., F. Prados, J. Poch, y J. Soler (2009). *La competencia «El trabajo colaborativo»: una oportunidad para incorporar las TIC en la didáctica universitaria. Descripción de la experiencia con la plataforma ACME (UdG)*. UOC Papers Revista Sobre la Sociedad del Conocimiento <http://uocpapers.uoc.edu>
- Esteve, F. (2009). *Bolonia y las TIC: de la docencia 1.0 al aprendizaje 2.0. La cuestión universitaria* (5), 59-68.
- Fallon, C. y S. Brown (2003). *e-Learning Standards. A Guide to Purchasing, Developing and Developing Standards-Conformant e-Learning*. USA, St. Lucie Press.

- Fernández-Manjón, B., J. L. Sierra, P. Moreno-Ger, e I. Martínez-Ortiz (2007). *Uso de estándares aplicados a TIC en Educación*. CNICE Centro Nacional de Investigación y Comunicación Educativa.
- Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York; Basingstoke: Palgrave Macmillan.
- Guin, C., V. Travers, B. Roberts, R. Tomlinson y D. Marvin (2008). *Supporting the Warfighter Training Community's Use of Game-Based Training. Proceedings of The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, Orlando, Florida (USA).
- Heller, R. S. (1990). *The role of hypermedia in education: A look at the research issues*. *Journal of Research on Computing in Education* 22 (4), 431-441.
- Hogarth, A. (2008). *Introducing a collaborative technology strategy for higher education students: Recommendations and the way forward*. *Educational and Information Technologies*. 13(3), 259-273.
- Hutchings, G. A., W. Hall, J. Briggs, N. V. Hammond, M. R. Kibby, C. McKnight y D. Riley (1992). *Authoring and evaluation of hypermedia for education Source*. *Computers & Education*, 18(1-3), 171-177.
- IMS Global Consortium (2003). *IMS Learning Design Specification, Version 1.0 Final Specification*. Retrieved 4<sup>th</sup> April 09 from <http://www.imsproject.org/learningdesign/index.html> .
- IMS Global Consortium. (2004). *IMS Content Packaging Specification, Version 1.1.4 Final Specification*. Retrieved October 7, 2008 from <http://www.imsglobal.org/content/packaging/index.html>.
- Kommers, P. y J. Lanzing (1997). *Students' concept mapping for hypermedia design: navigation through World Wide Web (WWW) space and self-assessment*. *Journal of Interactive Learning Research*, 8(3-4), 421-455.
- Manikonda, V., P. Maloor, J. Haynes y S. Marshall (2004). *An Architecture for Integrating SCORM-Compliant Instruction with HLA-Compliant Simulation; Proceedings of SISO Fall Simulation Interoperability Workshop, Orlando, Florida (USA)*.
- Malone, T. (1981). *Toward a theory of intrinsically motivating instruction*. *Cognitive Science* 5, 333-369.
- Malone, T., y M. Lepper (1987). *Making learning fun: A taxonomy of intrinsic motivations for learning*. *Aptitude, learning and instruction III: Cognitive and affective process analysis*, 223-253.
- McKnight, C., A. Dillon, y J. Richardson (1996). *User Centered Design of Hypertext and Hypermedia for Education*. In: D. Jonassen (ed) *Handbook of Research on Educational Communications and Technology*. New York: Macmillan, 622-633.
- Michael, D. y S. Chen (2006). *Serious Games: Games that Educate, Train, and Inform*. Boston, MA: Thomson.
- Moreno-Ger, P., D. Burgos, J. L. Sierra y B. Fernández-Manjón (2007a). *A Game-Based Adaptive Unit of Learning with IMS Learning Design and <e-Adventure>*. *Proceedings of the Second European Conference on Technology Enhanced Learning (EC-TEL 2007)*, Crete, Greece. *Lecture Notes in Computer Science* 4753, 247-261.
- Moreno-Ger, P., Sancho Thomas, P., Martínez-Ortiz, I., Sierra, J. L., & Fernández-Manjón, B. (2007b). *Adaptive Units of Learning and Educational Videogames*. *Journal of Interactive Media in Education*, 2007(05).

- Moreno-Ger, P., D. Burgos, J. L. Sierra, y B. Fernández-Manjón (2008).** *Educational Game Design for Online Education. Computers in Human Behavior.* **24(6), 2530-2540.**
- Moreno-Ger, P., D. Burgos y J. Torrente (2009).** *Digital Games in eLearning Environments Current Uses and Emerging Trends. Simulation & Gaming,* **In press (2009).**
- Ng, M. H., W. Hall, P. Maier y R. Armstrong (2002).** *The Application and Evaluation of Adaptive Hypermedia Techniques in Web-based Medical Education. Alt-J-Association for Learning Technology Journal,* **10 (3), 19-40.**
- Nussbaumer, A., C. Gütl, y C. Hockemeyer (2007).** *A Generic Solution Approach for Integrating Adaptivity into Web-based E-Learning Platforms. Proceedings of the International Conference on Interactive Mobile and Computer Aided Learning (IMCL 2007), Amman, Jordan.*
- Leutner, D. (1993).** *Guided discovery learning with computer-based simulation games: Effects of adaptive and non-adaptive instructional support. Learning and Instruction,* **3, 113-132.**
- Papert, S. (1998).** *Does Easy, Do It? Children, Games, Learning. Game Developer Magazine,* **87-88.**
- Paulsen, M. (2002).** *Online Education Systems: Discussion and Definition of Terms. NKID Education [www.nettskolen.com/in\\_english/web\\_edu.html](http://www.nettskolen.com/in_english/web_edu.html)*
- Prensky, M. (2001).** *Digital Game Based Learning. New York: McGraw-Hill.*
- Rey-López, M., R. P. Díaz-Redondo, A. Fernández-Vilas, J. J. Pazos-Arias, M. López-Nores, J. García-Duque, A. Gil-Solla y M. Ramos-Cabrer (2008).** *T-MAESTRO and its authoring tool: using adaptation to integrate entertainment into personalized t-learning. Multimedia Tools and Applications* **40(3), 409-451.**
- Torrente, J., P. Moreno-Ger, B. Fernández-Manjón y J. L. Sierra (2008).** *Instructor-oriented Authoring Tools for Educational Videogames. In proceedings of the ICALT 2008 conference. IEEE Computer Society, Santander, España. 516-518.*
- Torrente, J., P. Moreno-Ger, I. Martínez-Ortiz y B. Fernández-Manjón (2009).** *Integration and Deployment of Educational Games in e-Learning Environments: The Learning Objects Model Meets Educational Gaming. Educational Technology & Society, In press (2009).*
- Witthaus, G. (2009).** *The Implications of SCORM Conformance for Workplace e-Learning. Electronic Journal of e-Learning,* **2009, 7 (2), 183-190.**
- Yamada, S., J. Hong and S. Sugita (1995).** *Development and evaluation of hypermedia for museum education: validation of metrics. ACM Transactions on Computer-Human Interaction (TOCHI),* **2(4), 284-307, ACM.**



## 8. Anexos

### Anexo A. “*Bridging the Gap: Adaptive Games and Student-Centered VLEs*”

#### Referencia completa:

Ángel del Blanco, Javier Torrente, Pablo Moreno-Ger and Baltasar Fernández-Manjón: *Bridging the Gap: Adaptive Games and Student-Centered VLEs*; 4th International Conference on Web-based Learning (ICWL 2009), Aquisgrán, Alemania.

**Abstract.** The widespread e-learning technology is facing new challenges such as how to produce student-centered systems that can be adapted to the needs of each student. Those objectives should be met in a standard compliant way to simplify general adoption. In this context, educational videogames are proposed as an ideal medium to facilitate adaptation and tracking of the students’ performance for assessment purposes. However, there are still barriers between the gaming and e-learning worlds preventing their mutual interaction. In this paper we propose a middleware to bridge this gap, integrating adaptive educational videogames in e-learning environments with a special focus on the ongoing standardization efforts.

# Bridging the Gap: Adaptive Games and Student-Centered VLEs

Ángel del Blanco<sup>1</sup>, Javier Torrente<sup>1</sup>, Pablo Moreno-Ger<sup>1</sup>, Baltasar Fernández-Manjón<sup>1</sup>

<sup>1</sup> Universidad Complutense de Madrid. Facultad de Informática.  
C/ Profesor José García Santesmases s/n. 28040 Madrid, Spain  
{angel.dba, jtorrente, pablom, balta}@fdi.ucm.es

**Abstract.** The widespread e-learning technology is facing new challenges such as how to produce student-centered systems that can be adapted to the needs of each student. Those objectives should be met in a standard compliant way to simplify general adoption. In this context, educational videogames are proposed as an ideal medium to facilitate adaptation and tracking of the students' performance for assessment purposes. However, there are still barriers between the gaming and e-learning worlds preventing their mutual interaction. In this paper we propose a middleware to bridge this gap, integrating adaptive educational videogames in e-learning environments with a special focus on the ongoing standardization efforts.

**Keywords:** Educational videogames; Virtual Learning Environments; Adaptive Learning; Assessment; SCORM.

## 1 Introduction

Nowadays, the use of e-learning is increasing in industry and educational institutions. At the same time, e-learning systems have evolved from the original repositories of static content into richer Virtual Learning Environments (VLE), also known as Learning Management Systems (LMS), that follow different standards and specifications to assure the interoperability of the materials. The new student-centered features being adopted (such as VLE-driven adaptation of the learning experience or user tracking and assessment) demand further standardization efforts and raise new technical challenges.

Besides, there is an emergent trend in Technology-Enhanced Learning advocating for the use of educational videogames and game-like simulations [1, 2]. Educational videogames have key advantages such as their suitability to convey concepts [3] or to increase students' motivation [4]. Another key feature of educational games is that their high level of interactivity can provide fine-grained adaptation and performance-tracking mechanisms. This interactivity can open new possibilities in the e-learning field in terms of adaptive learning experiences, with more possibilities than those offered by more "traditional" hypermedia contents.

However, to bring the benefits of adaptive educational gaming to the e-learning field, we need to deal with the current diversity of VLE and with a lack of proper standardization support for the peculiarities of game-based learning.

This article presents a general architecture to integrate games in VLE with special emphasis on supporting adaptation and assessment. This architecture is designed to provide an abstraction layer (i.e. middleware) that allows game designers to create adaptive educational games without committing to a specific educational standard, thus offering the possibility of reusing the same game in different VLEs and contexts.

This paper is organized as follows. First, in section 2, we analyze the current state of the e-learning field in terms of assessment, adaptation and standards; then, in section 3 we discuss how videogames can contribute to assessment and adaptation in e-learning and the challenges behind this approach. Section 4 describes the architecture we propose and finally, in section 5, we present some conclusions and outline future lines of work.

## **2 VLEs: Assessment, Adaptation and Standards**

VLEs are rapidly evolving, giving the instructor more support and advanced tools to create complex online learning experiences. However, the increasing complexity of the content, including highly interactive pieces of content such as educational videogames, requires further support for the instructors. The new VLEs need to facilitate tasks such as tracking the progress and the skills acquired by the student within the games, as well as to adapt the learning experiences to the specific needs of each student.

Another important issue in e-learning is the interoperability of contents. At the present time, compliance to e-learning standards is a crucial factor when selecting a new VLE implantation within a corporate or educational environment.

E-learning specification and standardization initiatives are numerous and diverse, involving different organizations and consortiums such as the IEEE or IMS, and deal with the different aspects of the e-learning processes. This includes standardizing aspects such as supporting the creation of courses as aggregations of simple units of content (the Learning Objects Model [5]), the annotation of contents or the packaging of the courses. However, the ongoing standardization efforts are also covering more sophisticated aspects such as content sequencing or student profiling.

Even though the use of standards to package and distribute content is well established in the available VLEs, these standards do not simplify the widespread adoption of student-centered approaches with adaptation and performance tracking. For these reason, some initiatives such as the IMS Learning Design specification [6] are aiming to provide a standardized representation of the full learning process, taking into account pedagogical values.

With a lower degree of expressivity but a much wider adoption, the SCORM framework (Shareable Content Object Reference Model) has been proposed by ADL (Advanced Distributed Learning). This framework, an initiative of the U.S. administration to improve e-learning [7], is probably the most extended solution nowadays. In addition, SCORM defines a communication model that allows the

interchange of information between the (potentially interactive) content and the VLE in a standardized way.

However, currently two versions of SCORM coexist: SCORM1.2 and SCORM 2004 in most VLEs. Even though the newer version is more complete and adaptable, the full adoption of SCORM 2004 by mainstream VLEs (e.g. Moodle, Sakai or BlackBoard) is still an ongoing process. In addition, some environments such as those based on the IMS Learning Design specification [6] or supported by LAMS [8] are also valid alternatives for student-centered processes.

Therefore, the adoption of this type of advanced VLE demands dealing with a diversity of standards that may put the investment at risk. Given that developing interactive and adaptive content requires a significant budget, this can potentially become a major issue.

### **3 Game-based Learning and e-learning**

As it has been widely discussed in the literature during the last years [9], the use of videogames can enhance the learning processes in many aspects. The most frequently cited benefits of game-based learning are the increase in the motivation of the students, the relation between videogames and constructivist theories [4] or their support for collaborative/competitive learning. However, the full potential of videogames for education is almost undiscovered and requires further research.

#### **3.1 Videogames, adaptation and assessment**

Nowadays, personalization in videogames is common. Game developers and publishers include mechanisms in their videogames to adapt the game experience to suit the requirements of the widest possible range of users. The most obvious type of adaptation in videogames is the inclusion of different levels of difficulty, trying to adjust the challenge to different levels of skill.

However, the potential is even bigger thanks to the high interactivity of games, which can be used to implement much more fine-grained adaptation mechanisms, which can be carried out even transparently to the user. For example, the *MaxPayne*<sup>TM</sup> videogame incorporates Dynamic Difficulty Adjustment techniques [10] that alter the game execution depending on the actual performance of the user.

In addition, this fine-grained interactivity can also be monitored, allowing the collection of data gathered from user-game interaction that can be used to track the students' performance. This can bring new opportunities to the e-learning field to produce automatic assessment reports of the learning achievements of the player.

#### **3.2 Current challenges integrating game-based learning in VLE**

As previously discussed we believe that educational games can be an ideal medium to deliver student-centered content in VLEs. However, new challenges must be faced to

exploit successfully the potential synergies between adaptive game based learning and e-learning.

One issue concerns the flexibility and maintainability of the content, a key issue in e-learning but which is rarely present in videogames. In order to become a more useful tool in student-centered VLEs, the games must behave more openly, allowing the instructor to know what happens during the game sessions and to modify the behavior of the game. This requires the definition of specific models that allow the instructor to interact with the game experience remotely. This can be done using the currently existing standards mentioned in the previous section, but this approach presents two open issues.

On the one hand, a game developer who wants to integrate a game into a VLE must identify which standard/specification will be used in the VLE to store the data and how the games will exchange information with the VLE. Given the current situation, with diverse (and evolving) standards available, this does not guarantee the full interoperability of the contents, leaving the investment unprotected [11]. Besides, educational game developers must implement in each game the selected set of standards from scratch, which requires great efforts due to the inherent complexity of these standards. For a single videogame these issues may be acceptable, but if a large investment is planned, the problem becomes significant. This is especially relevant if we want to move towards educational experiences that contain diverse types of games that communicate with each other [2].

On the other hand, there are no tools that facilitate the addition of adaptive and tracking behavior to educational games. When developing an adaptive and assessable educational videogame, it is necessary to maintain a model of each student persistently, and developers must decide how to adapt the game experience according to that user model. If this behaviors are programmed ad-hoc in the game, the investment could become useless if instructors need to modify the adaptive and assessable behavior of the game (for instance, if the educational videogame is to be used in a new educational context). The problem of the student model persistence could be solved if instructors could directly set up the adaptation and assessment configuration of the game and connect the videogame with a LMS.

While the problem of connecting interactive content (such as games and interactive simulations) with LMS in standard-compliant ways has been partially addressed [12], there is still a need of research about how to use this connection automatically for adaptation and assessment purposes, and how to assure that the developed games will be resilient to future changes in the current standards.

Therefore, to facilitate the inclusion of educational games into the current student-centered VLE we need to achieve a greater independence between the implementation of the games and the standards used to connect them with the VLEs for adaptation and assessment.

## **4 The architecture**

In this section, we describe a general architecture that facilitates the integration of educational games in student-centered VLEs. The architecture was designed to

alleviate the potential issues described in the previous section in terms of standards compatibility, adaptation and assessment.

#### 4.1 Overview of the architecture

The architecture provides a two layer middleware. The first layer, called communication layer (CL), abstracts the communication between the game and the VLE. This layer is dependant on the specific standards supported by the VLE (e.g. SCORM). The VLE-specific data is translated to abstract adaptation and evaluation concepts. The abstract adaptation data is transmitted to the game-adaptation layer (GAL) that is specific for each game or game engine. The GAL translates this abstract adaptation constructs into specific actions or transformations within the game.

In addition, the GAL tracks the activity of the student and uses such information to produce an evaluation of the student in abstract terms. The assessment abstract commands are given to the CL, which sends it through a standard-compliant channel to the VLE. Thus, the CL adds independency from the specific standard supported by the VLE both for adaptation and assessment communication.

The fact that the middle-ware employs abstract adaptation and assessment constructs means that both the game and the VLE can be completely independent and interoperable.

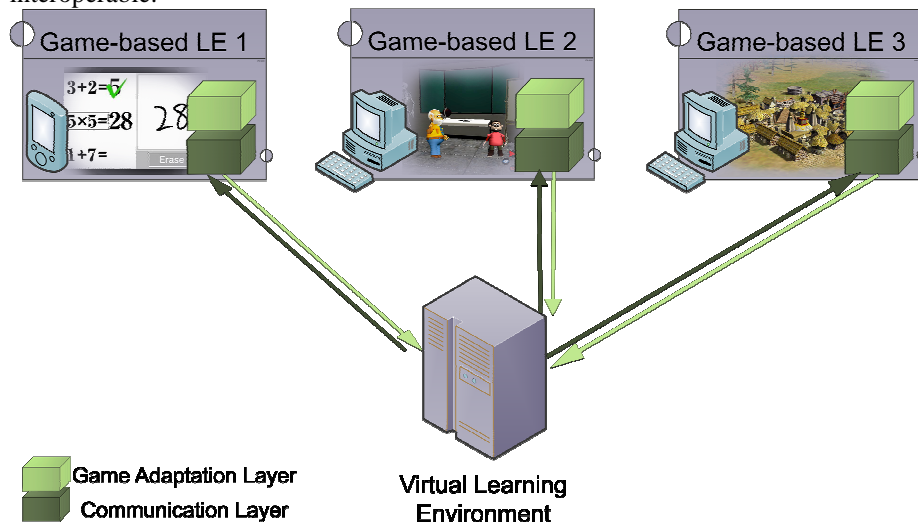
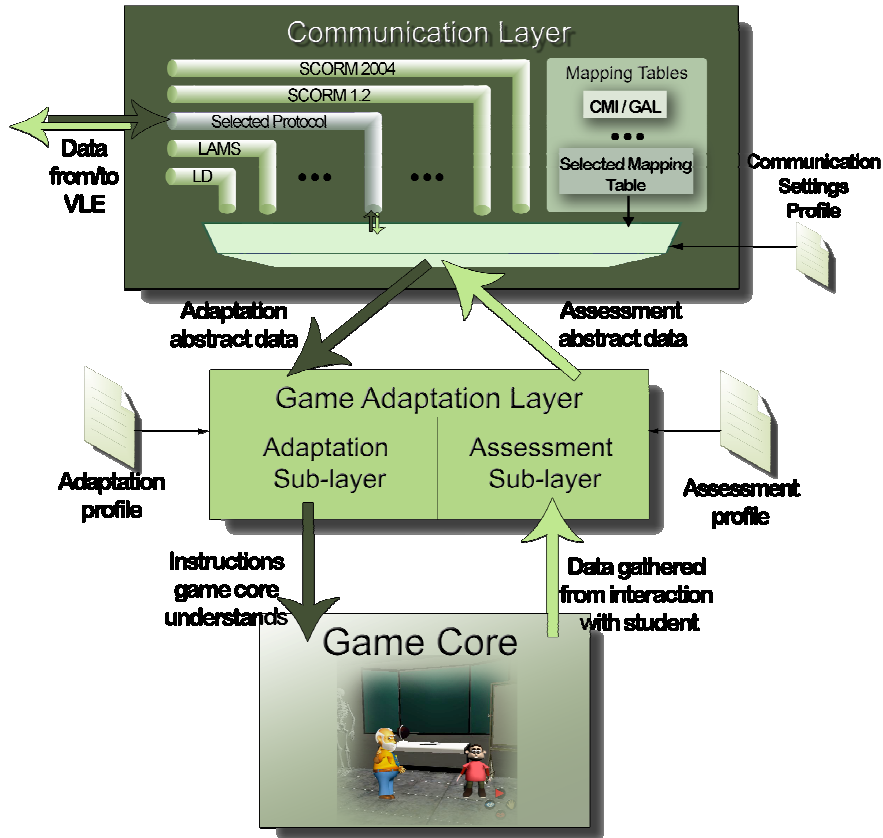


Fig. 1. General view of the architecture.

The role of the games in terms of assessment and adaptation is to provide valuable information about the performance of the student. This information is used by the middleware to drive a fine-grained online adaptation of the learning experience and to produce an assessment report that can be included in the VLE student profile. Those reports can be also used to update the student profile, therefore supporting a full adaptation cycle.

## 4.2 Using the double layer middleware

The CL layer includes implementations for some common standards in student-centered environments (based on specifications such as IMS-LD, SCORM1.2 or SCORM2004). When the game is run, the CL consults a configuration file (communication settings profile) to select the standard used to establish the communication with the VLE.



**Fig. 2.** Communication between the two layers of the middleware and the game core.

For each standard, the CL includes two relevant elements: an implementation of the communication protocol between the VLE and the game-based content that the VLE supports or defines, and a profile specifying how to map the abstract adaptation and assessment data model that the GAL understands to the communication data model defined by the standard. In some cases, the standards already provide a pre-established data model (e.g. CMI data model for SCORM). However, when the selected standard does not define the communication data model (e.g. Learning Design) the Communication Settings Profile would include the mapping table that the CL must use in the communication to store and retrieve the required information. In

these cases, the information contained in the profile will depend on the specific course (although it will be independent from the specific game).

The components of the CL are clearly isolated to allow flexible and systematic extensions and modifications of the layer, which guarantees interoperability and a longer life cycle for the architecture. New standards can be easily plugged in into the CL by carrying out two tasks: First, the communication protocol must be implemented following a specific API. In some cases this API will be provided by the standard (e.g. SCORM). In other cases, it will be dependant on the VLE. Secondly the mapping profile must be written, defining the translations between the abstract constructs and the VLE-dependant constructs.

In turn, the GAL focuses on translating the abstract constructs into game-specific commands. In a typical case, the GAL, via the adaptation sub layer (figure 2), polls the CL to get abstract adaptation commands (the CL gets the data from the LMS and translates it to the GAL language), and transforms those commands into game specific actions that modify the game experience. Then GAL will also listen to the interaction of the player in the game through an API that game developers must follow and will detect those situations that must trigger adaptation mechanisms. The specific actions that must be executed to adapt the game experience are defined by game authors through an adaptation profile (e.g. adjust the difficulty, give additional guidance, etc.).

The assessment sub-layer listens to the interaction between player and game as well. It uses also a profile (assessment profile), defined by the game author or the instructor, which determines the situations that must be assessed and how to track the activity of the student by defining assessment rules. When the game enters states matching any of the assessment rules (e.g. the student fails in completing a task) the assessment sub layer translates it into an abstract assessment construct.

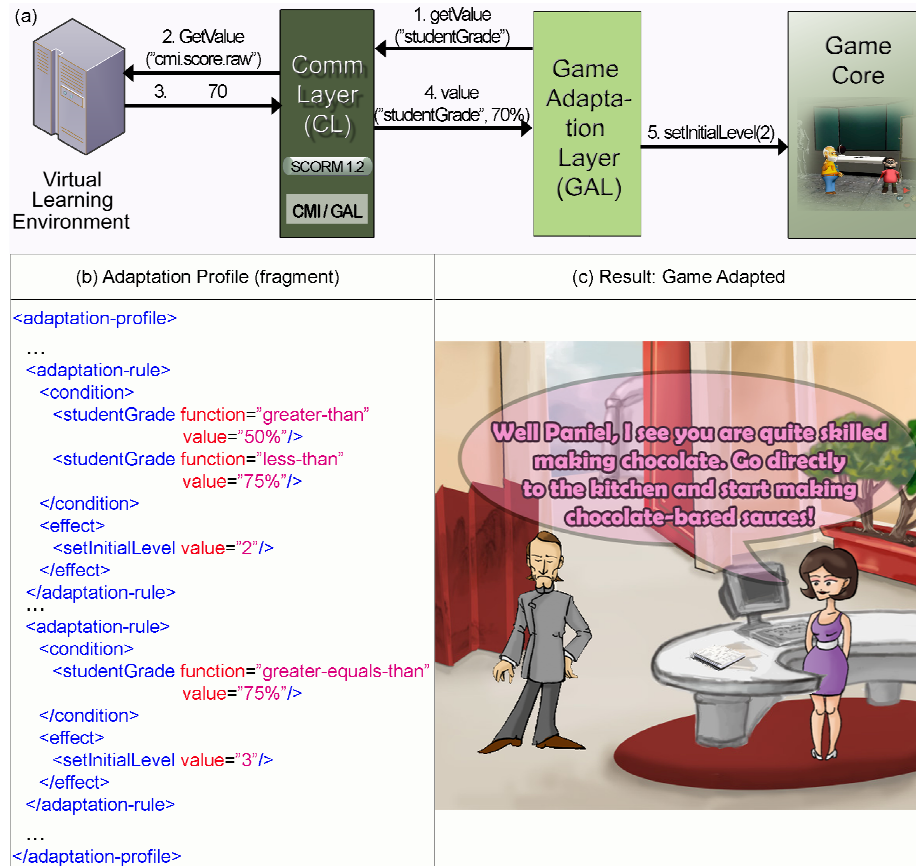
Then the CL transmits the information to the VLE, encoded according to the data models of the specific target environment, so that the VLE can attach the results to the profile of the student. In this manner the student is re-profiled and in future executions the adaptation can be performed more accurately.

With this structure the underlying communication standards, as well as the specific details of each game are completely transparent. Authors only need to modify the middleware configuration file to change the standards and specifications used in the communication. Besides, the double layer middleware helps instructors and educational game developers to think about adaptation and assessment more naturally, without considering technical communication details.

### **4.3. Example of the communication between VLE and game**

The architecture presented can be exemplified with the “*Paniel and the Chocolate-based Sauce Adventure*” game, which can be integrated into different student-centered VLEs. The game was originally developed to be integrated with a specific VLE which supports the IMS Learning Design specification [13]. The goal of the game is to introduce chocolate-based cooking techniques from a practical perspective, and is divided in three stages of different levels of difficulty. The initial level teaches

how make chocolate, the second level teach how make chocolate-based sauces, and the third level teaches how to combine them with dishes (the most challenging level).



**Fig. 3.** (a): Communication example diagram between the two layers, the VLE and the game core. (b): Fragment of the adaptation profile used in the example. (c): Screenshot of the game adapted.

For this game the GAL is set up with an adaptation profile that modifies the game depending on the prior knowledge of the student (figure 3b). When the game is executed, the GAL requests information from the VLE asking the overall grade of the student (figure 3a-step1). Then the CL, which has previously set up the communication channel with the VLE accordingly to the communication settings profile, codifies the request using the appropriate model, and sends it to the LMS. For instance, when deployed in a SCORM-compliant environment, the CL translates this abstract request into a check of the appropriate fields from SCORM's CMI data model (figure 3a-step2). The layer manages the request through the SCORM RTE and maps it to the GAL abstract data model (figure 3a-step4). The GAL then interprets the abstract response and uses the adaptation profile to decide to which level the student should go, skipping the first levels if appropriate. In this example, the adaptation

profile determines that if the student overall grade is greater than 50% and less than 75% the first level is skipped. If the overall grade is greater than 75% the second level is skipped. Finally, if due to any cause the overall grade could not be retrieved from the VLE, or if the overall grade is less than 50%, no levels are skipped (fig 3a-step 5).

Besides, the game includes an in-game test that produces a final grade. This abstract grade is translated to the appropriate fields of the SCORM data model and submitted to the VLE through the middle-ware, following the same steps (translation of the game concepts into abstract information and then translation into VLE-specific constructs). This grade can then be used in future executions of the game to make the initial adaptation decision.

## 5 Conclusions and future work

In this paper we have discussed the benefits that game-based learning can bring to e-learning in terms of adaptation and assessment. However, bringing both worlds together is a technically challenging task owing to the complexity of both fields. Especially relevant in this concern is the diversity of VLE communication and standards (SCORM, IMS LD, etc.) in e-learning.

Our contribution is a general architecture for the integration of games in VLE, consisting of a two-layer middleware which abstracts the existing standards. Using this architecture, the communication between a standards-compliant VLE and an adaptive educational videogame is independent of the specific game or standard. This allows game developers to create games without needing to be concerned with the internal details of each possible implementation of the student-centered VLE, focusing in this manner on the design of pedagogically relevant aspects. The interoperability, maintainability and reuse of the contents are addressed as the architecture is flexible enough to support new standards and revisions thanks to the notion of pluggable adapters.

We have tested this middleware in the <e-Adventure> educational game platform [14], which provides an authoring environment for educational games with special emphasis on the integration with VLEs [15], adding support for the APIs provided by the architecture. The preliminary results are promising, but also indicate some issues that will require further research.

On the one hand, adaptation is a complex issue. To exploit all the potential of adaptive game-based learning the abstract adaptation model must be extended and refined far beyond its current state. The discussion of how to adapt the content and in what circumstance it should be adapted is still an open research question. Moreover, the automatic detection of in-game situations which require adaptation deserves its own line of research.

On the other hand, the middleware must be expanded to include more modules for additional communication standards, including VLE-specific plug-ins for those environments that do not provide a standardized method for content-to-VLE communication. On the game side, we are also working on the implementation of the architecture for different game engines in different platforms.

**Acknowledgments.** The Spanish Committee of Science and Technology (projects FLEXO-TSI-020301-2008-19 and TIN2007-68125-C02-01) has partially supported this work, as well as the Complutense University of Madrid (research group 921340) and the EU Alfa project CID (II-0511-A).

## References

1. Tang, S., Hanneghan, M. and El Rhalibi, A.: Describing Games for Learning: Terms, Scope and Learning Approaches. The Fifth Annual International Conference in Computer Game Design and Technology, Liverpool, UK (2007) 98-102
2. Torrente, J., Lavín-Mera, P., Moreno-Ger, P., Fernández-Manjón, B.: Coordinating Heterogeneous Game-based Learning Approaches in Online Learning Environments. the Sixth International Game Design and Technology Workshop and Conference (GDTW2008), Liverpool, UK (2008) 27-36
3. Hamid, A.A.: e-Learning: Is it the "e" or the learning that matters? The Internet and Higher Education **4** (2001) 311-316
4. Gee, J.P.: What video games have to teach us about learning and literacy. Palgrave Macmillan, New York ; Basingstoke (2003)
5. Balatsoukas, P., Morris, A., O'Brien, A.: Learning Objects Update: Review and Critical Approach to Content Aggregation. Educational Technology & Society **11** (2008) 119-130
6. IMS Global Consortium: IMS Learning Design Specification, Version 1.0 Final Specification. (2003)
7. ADL: Advanced Distributed Learning Sharable Content Object Reference Model (ADL-SCORM). Vol. 2006 (2006)
8. Dalziel, J.: Implementing Learning Design: The Learning Activity Management System (LAMS). In: Crisp, G., Thiele, D., Scholten, I., Barker, S., Baron, J. (eds.): 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education. ASCILITE [www.ascilite.org.au](http://www.ascilite.org.au), Adelaide (2003)
9. Aldrich, C.: Learning by Doing: A Comprehensive Guide to Simulations, Computer Games, and Pedagogy in e-Learning and Other Educational Experiences. Pfeiffer, San Francisco, CA (2005)
10. Robin, H.: The case for dynamic difficulty adjustment in games. Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology. ACM, Valencia, Spain (2005)
11. Kanendran, T.A., Savarimuthu, J. and Durga-Kum, B.V.: e-Learning Standards. Proceedings of the 3rd European Conference on E-Learning, Paris, France (2004 ) 131-143
12. A. de Antonio Jiménez, R.R., J., & Madrigal Alfaro, Z.: Architecture for Integration of Simulations with SCORM in E-learning Environments. In proceedings of the ICALT 2008 conference, Santander, Spain (2008) 217-221
13. Burgos, D., P. Moreno-Ger, J.L. Sierra, B. Fernández-Manjón, M. Specht, & R. Koper: Building Adaptive Game-Based Learning Resources: The Marriage of IMS Learning Design and <e-Adventure>. Simulation & Gaming **39** (2008) 414-431
14. Torrente, J., Moreno-Ger, P., Fernández-Manjón, B., Sierra, J.L.: Instructor-oriented Authoring Tools for Educational Videogames. In proceedings of the ICALT 2008 conference. IEEE Computer Society, Santander, Spain (2008) 516-518
15. Moreno-Ger, P., Burgos, D., Sierra, J.L., Fernández-Manjón, B.: Educational Game Design for Online Education. Computers in Human Behavior **24** (2008) 2530-2540.



**Anexo B. “A General Architecture for the Integration of Educational Videogames in Standards-compliant Virtual Learning Environments”**

**Referencia completa:**

Ángel del Blanco, Javier Torrente, Pablo Moreno-Ger, Baltasar Fernández Manjón:  
***A General Architecture for the Integration of Educational Videogames in Standards-compliant Virtual Learning Environments***; 9th IEEE International Conference on Advanced Learning Technologies (ICALT 2009)

**Abstract.** Although Virtual Learning Environments (VLE) have become popular educational tools, they remain a very active research topic. Two important aspects of next-generation VLEs are how to track the performance of the students for assessment and how to provide personalized learning experiences (i.e. adaptive learning). In this line, educational videogames could contribute thanks to their highly interactive nature. However the combination of VLEs and educational games requires solving diverse conceptual and practical challenges. In this paper we present a general architecture to integrate games in VLEs with special emphasis on how to abstract the communication between the videogames and the VLEs for assessment and adaptation purposes.

# A General Architecture for the Integration of Educational Videogames in Standards-compliant Virtual Learning Environments

Ángel del Blanco<sup>1</sup>, Javier Torrente<sup>1</sup>, Pablo Moreno-Ger<sup>1</sup>, Baltasar Fernández-Manjón<sup>1</sup>

<sup>1</sup>Universidad Complutense de Madrid. Facultad de Informática.  
C/ Profesor José García Santesmases s/n. 28040 Madrid, Spain  
{angel.dba, jtorrente, pablom, balta}@fdi.ucm.es

## Abstract

*Although Virtual Learning Environments have become popular educational tools, they remain a very active research topic. Two important aspects being discussed for next-generation VLEs are how to track the performance of the students for assessment and how to provide personalized learning experiences (i.e. adaptive learning). In this line, educational videogames could contribute thanks to their highly interactive nature. However the combination of VLEs and educational games requires solving diverse conceptual and practical challenges. In this paper we present a general architecture to integrate games in VLEs with special emphasis on how to abstract the communication between the videogames and the VLEs for assessment and adaptation purposes.*

## 1. Introduction

Virtual Learning Environments (VLE), such as Moodle, Sakai, LRS or WebCT-Blackboard have become a powerful tool in education, providing instructors and students with multiple tools that support the learning process. In addition, VLEs are able to track the performance of the students not only through on-line exams, but also by monitoring the interaction between the student and the learning materials. VLEs can use this information to produce and maintain a persistent record of the activity of the student. Moreover, current research in VLEs explores how to use these student profiles to produce a user model of the student and personalize the learning experience according to that model. However, the effectiveness of VLEs for tracking the interaction is hindered by the passive nature of the web-based content, as it is very difficult to determine if html/hypermedia documents are really being attended or skimmed.

Another emerging trend is the use of videogames as educational materials due to their great educational potential, which has been thoroughly discussed in the

literature [1-3]. Among these benefits, one is especially relevant for this work: games are highly interactive, establishing a very short action-reaction feedback cycle with the player. This interactivity means that videogames can track students' actions to determine if the learning goals are being achieved [4]. In addition videogames can use the information gathered from user interaction to drive the on-line adaptation of the educational game experience [5]. However, videogames cannot keep a persistent record of the overall progress of the students in the course (which may include other forms of content and other games).

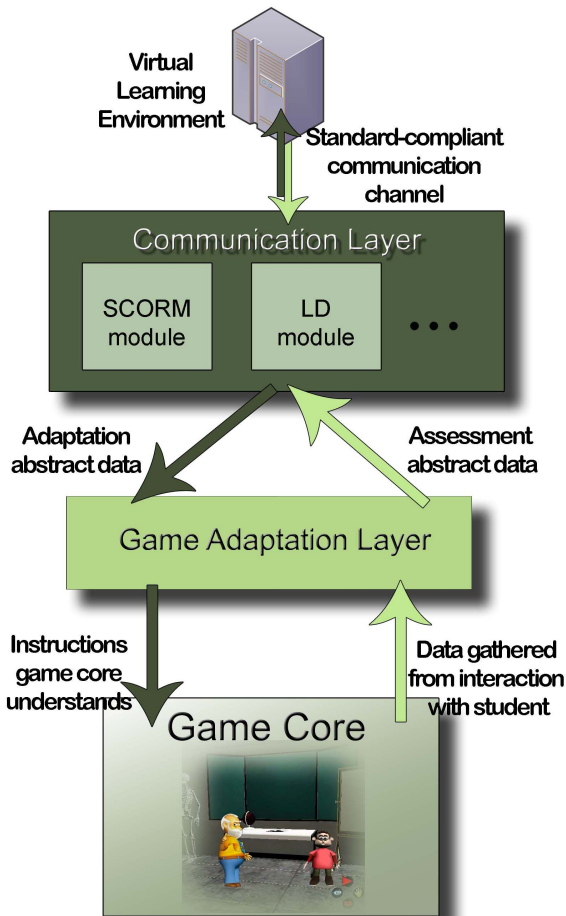
Therefore a synergy between VLEs and educational videogames could result in mutual benefit. On the one hand videogames can provide effective mechanisms to track the performance of the students. On the other hand VLEs can use that information to keep an updated profile of the student and use it for adaptation. Moreover, adaptation can be split in two stages: a first general adaptation phase directed by the VLE and a second, fine-grained in-game adaptation phase.

Nonetheless this integration poses significant challenges. Usually VLEs are ruled by standards and specifications that define the communication protocol between VLE and learning materials (e.g. SCORM [6]). If videogames are to be integrated in VLEs they must follow such rules, which is an extra technical burden for games authors, as many specifications coexist with no de-jure standard communication protocol. Furthermore, once the communication channel between VLE and videogame is established, the problem of how to use it for assessment and adaptation purposes arises, as these communication protocols were not designed to connect VLEs with so interactive content as videogames are [7].

In this paper we present a general architecture that aims to tackle both issues by abstracting the communication between the game and the VLE for adaptation and assessment purposes.

## 2. The architecture

The architecture behaves as a two-layer middleware that connects the VLEs with games hiding the technical details of the communication protocol. In addition the architecture provides an abstract adaptation and assessment data model (AADM) that aims to exploit the communication channel between VLE and game. The model is used to define the adaptive and assessable behavior of the game.



**Fig 1. General structure of the architecture**

The first layer is the Communication Layer (CL), which is responsible of establishing and managing the communication channel between the VLE and the game in a standards-compliant way. Thus the CL realizes actions such as *establish communication*, *get/set data from/to VLE* or *disconnect*. The CL decides the communication standard to be used (e.g. SCORM or an API provided directly by the VLE) and as a result these actions are dependent on the standards and specifications supported by the VLE. The CL needs to “understand” diverse specifications, which in this architecture is achieved through pluggable modules that implement a common API.

The CL uses the communication channel to provide services in terms of the AADM. Therefore the CL provides an API with methods such as *set student grade*, *send objectives results*, *get student name*, *get student skills level*, etc.

The second layer, which is called *Game Adaptation Layer (GAL)*, monitors the student-game interaction through an API that the game core must implement. The GAL uses the interaction data collected to maintain a record of the activity of the student. Besides the GAL uses the services provided by the CL to gather information from the VLE about the student, the course, etc. The activity record, along with the information provided by the VLE, can be used to personalize the game experience, even transparently to the student. For instance, the GAL could detect special situations such as if the student is lost in the game, if the difficulty level is too challenging/boring, etc., and perform some modifications in the game as defined by the instructor (e.g. *increment difficulty level*).

In addition, the record of the student can be used to evaluate the performance of the student. Instructors can identify situations that are relevant for student assessment (e.g. the game has not been completed successfully, a game task has been accomplished, etc.) and generate an assessment report (e.g. assign grade “A”, penalize with a 10% reduction of the grade, etc.). Then the GAL uses the CL to transmit the assessment report to the VLE, which processes the information and keeps it persistently in the student’s profile.

The behavior of the GAL is defined using the AADM. Thus the model supports the definition of both the situations that will trigger adaptation or assessment (e.g. *the student score is greater than 60%*) and the actions that must be performed in those situations (e.g. *set difficulty level “advanced”*).

## 3. Discussion of the architecture.

The architecture presented simplifies the integration of videogames into a VLE with numerous benefits. Educational game designers can focus on the design of the adaptive and assessable behavior of the games as they do not need to know the technical details of the communication. This adds not only persistence (the output of a game experience can be used as input for the next step in the course thanks to the VLE), but also a natural way to introduce games in complex courses just as any other learning resource.

However, some issues are still open. Probably the most relevant issue is that we must determine how the middleware is configured. This is complex due to the diversity of the standards that the CL must “understand”. Some of these standards define their own communication protocol and/or communication

data model (e.g. SCORM), but others do not (e.g. IMS Learning Design [8]). Hence the CL needs as input a *Communication Settings Profile* (CSP) which indicates the standard to use and, in some cases a definition of the communication protocol and/or data model and how to map it to the AADM.

In second place, the GAL must be configured as well, which includes a definition of how to carry out both adaptation and assessment. A valid solution could rely on *adaptation and assessment profiles* which would be basically a set of adaptation and assessment rules. Those rules could be compounded by a *condition* over the “state” of the game experience (which would include also information about the student profile that the VLE keeps) and a set of *effects*. Therefore when such conditions are satisfied the set of effects is triggered, which can both adapt the game experience or send data to the VLE to modify the student profile.

Here is where the strength of the architecture relies. The AADM should be enough flexible and abstract to allow instructors without a technical background to define how the game experience is to be assessed and adapted, which is a critical point from an instructional design perspective. Hence the AADM should help instructors to solve complex problems such as how to provide adaptation in terms of learning styles, prior knowledge, disabilities or even cultural background. Besides the adaptation and assessment model should include natural mechanisms to access the information stored in the profile of the student on the VLE side, supporting in this manner the two-stage adaptation process where VLE and game work together.

#### 4. Conclusions and future work

In this paper we present a general architecture to simplify the integration of videogames in VLEs. The architecture provides two layers (communication, adaptation) and is very flexible because it can be configured to work with multiple VLEs and videogames. Support for different communication standards can be easily added by defining pluggable modules in the communication layer. Finally, the architecture supports the collaboration of multiple roles in the definition of game-based courses, as it isolates the tasks that game developers, VLE administrators and instructors must carry out.

This work is mainly a contribution from a Software Engineering perspective, and we have also implemented and tested several of the ideas discussed here. On the one hand, the implementation of the CL is complete, including communication modules for SCORM (version 1.2) and VLEs based on .LRN supporting IMS Learning Design. On the game side, we have tested the architecture with the <e-Adventure>

[9] educational game platform. This platform supports the concepts of assessment, adaptation, and communication settings profiles, and provides utilities to edit them when designing an educational game.

Therefore, our next work will be focused on modelling the AADM, which is a great challenge, and release a development framework that implements the principles of the architecture.

#### Acknowledgments

The Spanish Committee of Science and Technology (projects FLEXO-TSI-020301-2008-19 and TIN2007-68125-C02-01) has partially supported this work, as well as the Complutense University of Madrid (research group 921340) and the EU Alfa project CID (II-0511-A).

#### References

- [1] D. Michael and S. Chen, *Serious Games: Games that Educate, Train, and Inform*. Boston, MA: Thomson, 2006.
- [2] J. P. Gee, *What video games have to teach us about learning and literacy*. New York ; Basingstoke: Palgrave Macmillan, 2003.
- [3] C. Aldrich, *Learning by Doing: A Comprehensive Guide to Simulations, Computer Games, and Pedagogy in e-Learning and Other Educational Experiences*. San Francisco, CA: Pfeiffer, 2005.
- [4] P. Moreno-Ger, D. Burgos, J. L. Sierra, and B. Fernández-Manjón, "Educational Game Design for Online Education," *Computers in Human Behavior*, vol. 24, pp. 2530-2540, 2008.
- [5] P. Moreno-Ger, P. Sancho Thomas, I. Martínez-Ortiz, J. L. Sierra, and B. Fernández-Manjón, "Adaptive Units of Learning and Educational Videogames," *Journal of Interactive Media in Education*, vol. 2007,
- [6] ADL, "Advanced Distributed Learning Sharable Content Object Reference Model (ADL-SCORM)." 2006, Retrieved 4<sup>th</sup> April 09 from [www.adlnet.gov](http://www.adlnet.gov).
- [7] R. R. A. de Antonio Jiménez, J., & Madrigal Alfaro, Z., "Architecture for Integration of Simulations with SCORM in E-learning Environments," in *Proceedings of ICALT 2008*, Santander, Spain, 2008, pp. 217-221.
- [8] IMS Global Consortium, "IMS Learning Design Specification, Version 1.0 Final Specification," 2003. Retrieved 4<sup>th</sup> April 09 from: <http://www.imsproject.org/learningdesign/index.html>
- [9] J. Torrente, P. Moreno-Ger, B. Fernández-Manjón, and J. L. Sierra, "Instructor-oriented Authoring Tools for Educational Videogames," in *Proceedings of ICALT 2008*, Santander, Spain, 2008, pp. 516-518.