

Estado del arte del eLearning. Ideas para la definición de una plataforma universal.

José Manuel Márquez Vázquez

DNI. Nº 28764391-Q

Supervisado por el Prof. Dr. Juan Antonio Ortega Ramírez



Departamento de

Lenguajes y Sistemas Informáticos

Universidad de Sevilla

Informe de investigación enviado al Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla en cumplimiento parcial de los requisitos para la superación del Período Investigador del Doctorado en Lenguajes y Sistemas Informáticos.

Sevilla, marzo de 2007

“Todos los que se interesen sinceramente por los problemas de la educación y de la comunicación y que son conscientes de su interdependencia tratarán durante los próximos años de establecer los vínculos entre esos dos informes y de garantizar una simbiosis práctica, operativa, entre las dos culturas, la escolar y la de los medios de comunicación, cada una en su especificidad y sin prelación.”

UNESCO, 1984.

Índice

1. Introducción.....	11
1.1. Educación y comunicación.....	11
1.2. Reciente evolución de los procesos de enseñanza a distancia	12
1.3. Las Tecnologías de la información en la enseñanza.....	14
1.3.1. Las TIC y la enseñanza secundaria en España.....	15
1.3.2. Infraestructura y acceso a las tecnologías de la información.....	18
2. Estado del arte	20
2.1. Términos asociados al eLearning.....	20
2.2. Ventajas del eLearning.....	20
2.3. Inconvenientes del eLearning.....	21
2.4. ¿Por qué usar eLearning?.....	22
2.5. Especificaciones y estándares en eLearning.....	23
AICC.....	26
ADL.....	27
IMS.....	30
IEEE LTSC.....	42
CEN/ISSS.....	45
Resumen.....	46
2.5.1. Relación entre especificaciones y procesos de aprendizaje.....	47
Evolución de especificaciones relacionadas con los contenidos.....	47
Evolución de especificaciones relacionadas con la funcionalidad del LMS.....	49
Evolución de especificaciones relacionadas con la metodología.....	50
2.6. Herramientas.....	51
2.7. La plataforma tecnológica de eLearning.....	66
2.7.1. Arquitecturas de los sistemas de enseñanza a distancia.....	67
3. Planteamiento del problema	71
3.1. Introducción.....	71
3.2. Relevancia.....	72
3.2.1. Desde el punto de vista económico.....	73
3.2.2. Desde el punto de vista legal.....	73
3.2.3. Desde el punto de vista social.....	74
3.2.4. Desde el punto de vista tecnológico.....	74
3.2.5. Desde el punto de vista de la enseñanza.....	74
3.2.6. Desde el punto de vista de la investigación.....	75
3.3. Problemas existentes.....	76
3.4. Definición del problema.....	77
3.5. Propuesta de solución	79
3.5.1. Despliegue de componentes	80
Centro de Control.....	84
Plataforma OSGi.....	85
Modelo de aplicación y descriptores de despliegue.....	85
Descriptor de J-Bones.....	86
Descriptor OAM.....	87
El proceso de despliegue.....	88
3.5.2. Integración de dispositivos y generación de eventos nativos de bajo nivel.....	89
3.5.3. Taxonomía de las aplicaciones.....	89
Creación de aplicaciones Desktop-Friendly.....	93
3.5.4. Técnicas de adaptabilidad.....	97
3.6. Justificación del proyecto.....	98
4. Proyectos.....	100
4.1. EDUKA.....	100
4.2. OSMOSE.....	101

4.3. Passepartout.....	112
5. Conclusiones.....	129
5.1. Trabajo futuro.....	130
6. Curriculum Vitae	131
7. Referencias bibliográficas.....	134
8. Foros relacionados.....	137
8.1. Foros y portales en internet.....	137
8.2. Revistas.....	138

Índice de figuras

Figura 1. Especialización en la creación de cursos eLearning.....	22
Figura 2. Entidades participantes en estandarización de especificaciones.....	24
Figura 3. Proceso de estandarización.....	24
Figura 4. Libros que componen la especificación de SCORM.....	29
Figura 5. Diagrama ilustrativo de SCORM CAM.....	30
Figura 6. Estructura del Manifiesto.....	32
Figura 7. Estructura del fichero imsmanifest.xml.....	33
Figura 8. Contenido empaquetado en un fichero .zip.....	35
Figura 9. Modelo conceptual de IMS Learning Design	38
Figura 10. Estructura de un paquete de contenidos IMS CP	39
Figura 11. Estructura de una UoL	39
Figura 12. Learning Technology Systems Achitecture (IEEE 1484.1)	44
Figura 13. Evolución de especificaciones sobre contenidos en eLearning.....	48
Figura 14. Modelo funcional del Sistema eLearning de Liu, El Saddik y Georganas	67
Figura 15. LTSA presente en el modelo funcional de Liu, El Saddik y Georganas.....	68
Figura 16. Arquitecturas Orientadas a Servicios.....	69
Figura 17. Semejanzas en el empaquetado de cursos y bundles OSGi.....	81
Figura 18. Grafo de dependencias a nivel de paquetes	82
Figura 19. Manifiesto de un bundle.....	82
Figura 20. Grafo de dependencias funcionales entre bundles.....	83
Figura 21. Arquitectura del Centro de Control.....	84
Figura 22. Arquitectura de la plataforma de servicios OSGi.....	85
Figura 23. Modelo de aplicaciones.....	86
Figura 24. Asociación de bundles y aplicaciones.....	88
Figura 25. Modelo de aplicaciones con interfaz gráfica de usuarios.....	90
Figura 26. Gestión de aplicaciones finales de usuario por GUIService.....	91
Figura 27. Interfaz DesktopInterface.....	92
Figura 28. Interfaz DesktopManagerInterface.....	92
Figura 29. Escenario de OSMOSE: el hogar inteligente.....	101
Figura 30. Arquitectura básica de una pasarela de servicios OSGi.....	103
Figura 31. Esquema de modelo de dependencias entre aplicaciones.....	105
Figura 32. Componentes del mecanismo de despliegue.....	105
Figura 33. Comunicación de AxisBundle con el resto de componentes de la pasarela.....	108
Figura 34. Aplicación del Patrón Observador en el diseño del Escritorio.....	110
Figura 35. Servicio GUIService y su relación con el Escritorio.....	110
Figura 36. Patrón Comando.....	111
Figura 37. Apariencia externa final de la pasarela.....	112
Figura 38. Arquitectura y tecnologías usadas en Passepartout.....	113
Figura 39. Soporte para control remoto por infrarrojos en el escritorio.....	115
Figura 40. Esquema de adaptación de contenidos en tLearning.....	116
Figura 41. Módulos funcionales de tMAESTRO.....	117
Figura 42. Inclusión del fichero de reglas de adaptación en un SCO.....	118
Figura 43. Escritorio: menú y accesos directos (iconos).....	121
Figura 44. Control Panel: Gestor de bundles desplegados en la plataforma.....	121
Figura 45. Control Panel: Acceso al Oscar Bundle Repository para obtener nuevas utilidades.....	122
Figura 46. Control Panel: Utilidad Shell que permite ejecutar comandos del framework.	122
Figura 47. Selección de usuario (Ana).....	123
Figura 48. Recomendación de cursos para Ana en la aplicación tMAESTRO.....	123
Figura 49. Primera pantalla del curso sobre la Unión Europea.....	124
Figura 50. Navegación con el control remoto en el guión del curso SCORM.....	124
Figura 51. Selección del primer punto del primer capítulo: History – A little journey.....	125
Figura 52. Visualización del recurso asociado al SCO (asset) seleccionado (I).....	125
Figura 53. Visualización del recurso asociado al SCO (asset) seleccionado (II).....	126
Figura 54. Selección de usuario (Pedro).....	126

Figura 55. Recomendación de cursos para Pedro en la aplicación tMAESTRO.....	127
Figura 56. Visualización del recurso asociado al SCO.	127
Figura 57. Votación de contenidos: Pedro otorga un 8 a “Telediario 2”.....	128

Índice de Tablas

Tabla 1. Principales organizaciones en la elaboración de especificaciones y estándares en eLearning.....	25
Tabla 2. Otras organizaciones que ha participado directa o indirectamente en el desarrollo de estándares de eLearning.....	25
Tabla 3. Algunos acuerdos producidos por CEN/ISSS.....	46
Tabla 4. Clasificación de especificaciones de eLearning según área.....	46
Tabla 5. Lista de webs observatorio del estado del eLearning en España.....	66

Índice de Listados

Listado 1. Ejemplo de Manifiesto IMS CP versión 1.1.2.....	35
Listado 2. Ejemplo de fichero descriptor de OAM.....	88
Listado 3. Ejemplo de Manifiesto de un bundle.....	103
Listado 4. Interfaz BundleActivator.....	104
Listado 5. Manifiesto del bundle desktop.jar con que depende del anterior.....	104
Listado 6. Métodos de la interfaz AxisBundleFunctionality.....	107
Listado 7. Ejemplo de fichero adaptation.xml.....	119

Agradecimientos

Quiero dar las gracias, en primer lugar, a mi tutor en el período investigador, que ha confiado en mí y me ha apoyado en todo a pesar de las dificultades, lo que me ha servido de aliento para llevar a cabo este trabajo. Gracias Juan Antonio.

También he de agradecer a los compañeros de trabajo que he tenido en estos años en TELVENT. Su comprensión en mis momentos de agobio y la ayuda que me han prestado cuando lo he necesitado ha sido de inestimable valor. Gracias especialmente a Jesús González por su valiosa ayuda en el tramo final, a José Antonio Alonso de quién tanto aprendí y a Guillermo Hernández por su colaboración. Gracias también a Manuel Muñoz, Fernando Usero, Miguel Ángel Oltra, Javier Álamo, Jesús Bermejo...

Quiero agradecer también a los “compañeros ITEA” por todos esos buenos momentos que hemos pasado juntos debatiendo acerca de OSGi, eLearning, y “tantos otros temas” de los que siempre se aprende: José Luis Ruiz, Juan Carlos Dueñas, Manuel Santillán, Alberto Gil, Carlos Acuña, Miguel García Longarón, John Pleunis, Rop Pulles, Rick Hall, Thor Erland Faegri, Naci Day, Svein Hallsteinsen y Martin Treiber.

Gracias a los miembros de la lista de distribución ELEARN de RedIris, y especialmente a su moderador, David Puente, por su colaboración en la búsqueda de recursos de información y sus valiosas opiniones.

Y gracias a Isabel por su paciencia, comprensión y estímulo.

Resumen

En esta memoria se presenta el trabajo de investigación sobre especificaciones y estándares de eLearning y una propuesta de integración de herramientas de eLearning en pasarelas residenciales basadas en la especificación de OSGi, que creemos ayudará a la definición de una plataforma universal, abierta, escalable, útil tanto para el seguimiento de cursos ofrecidos por un proveedor remoto como para la reproducción de cursos desplegados sobre dicha plataforma (empaquetados según las especificaciones de eLearning estudiadas) mediante un gestor de aprendizaje (LMS) local. De esta forma el usuario podrá seguir cursos on-line, valiéndose de un entorno colaborativo con el que interactuar con sus compañeros y profesores, u off-line, mientras el usuario se desplaza (por ejemplo en avión) o no dispone de conexión a Internet. No es el propósito de este trabajo convencer de que se trata de la mejor solución o alternativa posible, ni justificar su elección frente a otras alternativas, es simplemente un intento de aplicación de OSGi al campo del eLearning, o visto de otra forma, organización de herramientas y aplicaciones de eLearning para poder ejecutarse sobre un framework OSGi.

En el estudio del arte se analizan las características de diferentes gestores del proceso de aprendizaje, cuyo denominador principal es que todos ellos son aplicaciones open-source, no entrando en el ámbito de este trabajo aquellas herramientas comerciales o que no proporcionen el código fuente. También se aporta en esta memoria un conjunto ideas para la incorporación de técnicas encaminadas a la adaptación tanto de dichos contenidos como del entorno de ejecución de éstos a las necesidades del usuario.

Esta memoria de investigación se ve reforzada por los avances obtenidos en dos proyectos de investigación europeos en los que el autor ha participado, y de cuyo trabajo se aporta un breve resumen.

1.Introducción

1.1.Educación y comunicación

El avance de la ciencia ha influido desde siempre en la sociedad, pero en las últimas décadas se ha intensificado notablemente, llegando el ser humano a cuestionarse su dependencia de tales avances tecnológicos.

Podríamos decir que desde que el hombre se comunica con sus semejantes y posteriormente fue capaz de estructurar un lenguaje verbal oral, y sobre todo escrito, ha visto la necesidad de establecer algún mecanismo de acción que permitiera la conversación y transmisión de sus saberes. Al hombre no le bastaba con la comunicación en sí, y creyó necesaria la transmisión de sus conocimientos a su prole, sus congéneres... Nació así la educación en su estado más primitivo: la intención de establecer un mecanismo de comunicación mediante el cual transmitir una serie de conocimientos.

La educación se adquiere a través de procesos comunicativos. Pero esto no implica que todos los procesos comunicativos sean educativos. Para ello es necesario el aporte de un beneficio para el desarrollo de la persona, mejora, perfeccionamiento. En definitiva una adquisición de conocimientos, ya sean nuevos conceptos, habilidades, técnicas o actitudes y modos de comportamiento.

Pero ¿cómo podemos diferenciar procesos meramente comunicativos de los procesos educativos? Por la intencionalidad presente en la educación. En la comunicación, la intención, el objetivo del emisor del mensaje es persuadir, informar, entretener, formar, pero puede que al receptor no le interese el contenido de ese mensaje, puesto que no siempre la intencionalidad del emisor coincide con la del receptor. De hecho, en numerosos casos se discute abiertamente o recibimos mensajes que no nos interesan en absoluto. No ocurre o no debería ocurrir en los procesos educativos, donde el propósito del docente es educar, lo cual no puede llevarnos a equívocos; existe una intencionalidad y una voluntariedad por parte del docente en enseñar y educar y por parte del alumno o discente en ser educado o aprender. Ésta sería, a nuestro modo de entender, la gran diferencia entre comunicar y educar.

La tarea de los formadores es presentar, dirigir, discriminar, seleccionar, ordenar, orientar esos procesos comunicativos para convertirlos en procesos educativos. Según cómo el formador realice estas actividades, es posible identificar varias metodologías de enseñanza: conductista, constructivista, etc.

Cuando los procesos educativos se manifiestan en la práctica, de forma real, se habla de situaciones de enseñanza. Y según dónde y cuando se produzcan estas situaciones, tendremos:

Enseñanza presencial: uno y otro, docente y discente, están presentes en espacio y tiempo, cara a cara, en el mismo recinto espacial y temporal

Enseñanza a distancia: uno y otro están separados por tiempo y por distancia. Una particularidad de la enseñanza a distancia, es la enseñanza on-line, en la que gracias al uso de las tecnologías de la información y comunicación (correo electrónico, teléfono, TV, vídeo, ordenador o redes informáticas), se establece un contacto mediatizado, que hace disminuir enormemente el tiempo que dista entre uno y otro, facilitando la interacción hasta de forma virtual en menor o mayor medida.

Enseñaza semipresencial. La *enseñanza a distancia* es un concepto en permanente a evolución, que ha evolucionado desde una “distancia total” a formas combinadas de presencia y ausencia. Lo mismo ha pasado con la enseñanza on-line o eLearning, porque la interacción cara a cara entre docentes y alumnos, así como entre los propios alumnos, permite llevar a cabo tareas que no son viables a distancia [1] especialmente aquellas con una marcada componente motriz. Se han ido incorporando progresivamente a la enseñanza presencial elementos que provenían de la experiencia en la enseñanza a distancia, como por ejemplo, el uso más intenso de materiales educativos multimedia. Hay autores que hacen referencia a la “webización” de la enseñanza [2] cuando se utiliza Internet como herramienta de apoyo a la educación presencial dado el alto porcentaje de profesores que recurren a la web para buscar recursos para sus clases: el 82.7% de los profesores en la Europa de los 25 y el 84.26% de los profesores españoles usan material de Internet como apoyo para su asignatura, según el informe de la Comisión Europea [3] Suele hablarse de *blended learning* o educación semipresencial, para referirse a esos procesos en los que se conjugan procesos educativos que requieren la presencia del alumno y procesos que no la requieren. Algunos autores han acotado este confuso intervalo de presencia entre el 25 y el 75 por ciento del total del tiempo lectivo [4] para la enseñanza semipresencial, identificando así a los que requieran un tiempo inferior al 25% como enseñanza a distancia y a los que requieran más de tres cuartas partes del total lectivo como enseñanza presencial.

1.2. Reciente evolución de los procesos de enseñanza a distancia

La evolución en la educación y la formación a distancia se ha ido especializando según el medio de comunicación empleado. A esta especialización han contribuido destacadamente las nuevas tecnologías de la información, influencia que se ha dejado notar en el término con el que se distinguen cada una de estas especializaciones, pasando así de un genérico *d-Learning* (del inglés, *distance learning*, aprendizaje a distancia) a lo que actualmente llamamos *eLearning* (del inglés, *electronic Learning*, aprendizaje electrónico) y recientemente a raíz de éste *m-Learning* (del inglés, *mobile Learning*, aprendizaje móvil) y t-learning (para aquellos procesos que usan la televisión, especialmente la televisión digital, como medio de comunicación). Como se puede deducir de los términos anglosajones, cada uno de estos términos va asociado a la naturaleza del soporte y medio por el que transporta la información desde el docente al discente. Estos estados de desarrollo corresponden a la influencia que ha tenido en la sociedad cada una de las siguientes *revoluciones*: la Revolución Industrial en los siglos XVIII y XIX, la Revolución Electrónica en la década 1980-1990, y la Reciente Revolución Inalámbrica comenzada en los últimos años del siglo XX.

La Revolución Industrial

La educación y formación a distancia nació en el norte de Europa y América a finales del siglo XVIII y comienzos del XIX. No fue un accidente que la enseñanza a distancia comenzase con el desarrollo de tecnologías industriales, especialmente la comunicación postal y el transporte. Incluso hoy, la formación a distancia no sería posible en una sociedad que no hubiera adquirido un cierto nivel de industrialización.

La Revolución Electrónica

La industria de las telecomunicaciones experimentó grandes y complejos cambios a principio de los años ochenta, constituyendo lo que se ha dado en llamar la Revolución Electrónica o Revolución del Chip.

Antes de la Revolución Electrónica, los gobiernos concebían las telecomunicaciones como un monopolio industrial altamente lucrativo, y en general altamente ligado a instalaciones de defensa militar. Había una total regulación.

Sin embargo, en el Reino Unido y Estados Unidos, bajos los gobiernos de Thatcher y Reagan respectivamente, se rompieron estos monopolios y las telecos comenzaron a convertirse en un buen indicador de consumo. La informática se introdujo por primera vez en las telecomunicaciones en los años sesenta, para digitalizar las centralitas telefónicas. Ya en los años ochenta se introdujo la Red Digital de Servicios Integrados (RDSI). En los noventa, se posibilitó la interconexión entre redes de cable y por aire. En todos estos avances, la cada vez mayor frecuencia de trabajo de los chips fue crucial. El proceso se aceleró aún más cuando se remplazaron los chips de silicio por los diminutos nanochips.

El desarrollo de la tecnología de banda ancha es de vital importancia para la formación a distancia, dado que es necesario transmitir simultáneamente contenidos que requieren una alta velocidad de transferencia de datos como: imágenes, audio, vídeo, contenidos interactivos multimedia y realidad virtual. El término Banda Ancha, se suele utilizar generalmente para conexiones que soportan una tasa de transferencia superior a 2 Mbits por segundo.

La Revolución Electrónica de los ochenta ha abierto el camino de la web a la formación a distancia, minimizando increíblemente el tiempo necesario para acceder a la información buscada.

La Revolución Inalámbrica

Al final de 1999 la población del mundo alcanzó los seis billones de personas por primera vez. Al mismo tiempo, Ericsson y Nokia anunciaban que había 500 millones de teléfonos móviles en el mundo y que serían un billón en el año 2004.

La Revolución Electrónica de los ochenta modificó la naturaleza de la educación a distancia, haciendo posible la enseñanza cara a cara de forma virtual, así como enseñar a grupos y a personas individuales de la misma forma a través de la videoconferencia. La aún joven Revolución Inalámbrica cambiará el concepto de educación a distancia no sólo ya por conceder al alumno la elección de asistir o no al centro académico, sino que además facilitará el acceso a la información y a procesos educativos, al mismo tiempo que se desplaza por el mundo.

Los sistemas de tele-enseñanza se están implantando lentamente y aunque algunas especificaciones comienzan a ser aceptadas convirtiéndose en estándares de facto, aún sigue existiendo una gran heterogeneidad en cuanto a tipos de sistemas de gestión del aprendizaje. La mayoría de estos sistemas son *en-línea* y requieren de una conexión permanente a Internet, requisito que no puede satisfacerse siempre en dispositivos móviles, que dependen de la cobertura existente en un área o, en el peor de los casos, ni siquiera ofrecen la posibilidad de conexión a la red. Sin embargo hay una razón para el optimismo: La telefonía móvil es la tecnología que más impacto ha tenido en la sociedad en los últimos años, y aunque el mercado de los PDA se ha estabilizado en los últimos años, su crecimiento y nivel de ventas también es un factor positivo. Las estadísticas de la disponibilidad de teléfonos móviles es un fuerte indicador de la necesidad de desarrollar estrategias y estructuras didácticas de m-Learning.

A lo largo de la historia, los sistemas que han subsistido han sido los que han gozado de una mayor accesibilidad para el público en general. Tecnologías con increíbles facilidades didácticas como el Láser-Disc, a principios de los noventa, no tuvieron éxito porque no eran accesibles para los estudiantes y pocos podían disfrutar de ellos en sus hogares.

Las estadísticas de Ericsson a mediados de 2001 mostraban que China, a pesar de su régimen comunista, ya era el país que más teléfonos móviles tenía, con 170 millones, por encima de Japón y Estados Unidos, lo que nos da una idea de la alta accesibilidad de esta tecnología.

Otros indicadores nos hacen pensar en los dispositivos móviles (telefonía móvil, PDAs, tablet PC...) como plataformas para combinar aprendizaje y movilidad:

- Más del 75% de los contenidos mostrados en Internet en 2002 puede ser portado fácilmente a dispositivos móviles.

- El número de dispositivos móviles ya superó en diciembre de 2002 al número de PCs.
- El comercio a través de dispositivos móviles en 2004 superó los 200 billones de euros.
- Se prevé que haya más de un billón de suscripciones a Internet a través de redes inalámbricas.
- Más del 50% de empleados pasan la mitad de su jornada fuera de la oficina.

Sin embargo, el alto impacto de la telefonía móvil y las redes de comunicaciones, en especial Internet, en la sociedad no son suficientes para garantizar la idoneidad de cualquier sistema de enseñanza a distancia basado en estas tecnologías. Depende también del nivel educativo y perfil psicológico del alumnado al que se quieran aplicar y que pasamos a analizar más detenidamente a continuación

1.3.Las Tecnologías de la información en la enseñanza

Por tecnologías de la información y del conocimiento (TIC) suele entenderse el conjunto de herramientas, soportes y canales para el tratamiento y acceso a la información, que generan nuevos formatos de expresión, nuevas formas de acceso y nuevos hábitos en definitiva de conducta social, cultural y del ocio. Para algunos analistas los cambios sociales y las nuevas estructuras socioeconómicas responden a las innovaciones técnicas (determinismo tecnológico). Pero si bien la tecnología tiene un gran impacto en la configuración de las relaciones sociales, la economía o la política que caracteriza a la sociedad actual, también es cierto que la propia sociedad ha influenciado a raíz de un doble proceso de base social para que dichas innovaciones técnicas aparezcan: de un lado las economías de los países desarrollados que a partir de la crisis de los setenta exigían una completa flexibilización de sus procesos productivos y la globalización de sus actividades. Y de otro lado, porque estas sociedades avanzadas están presididas por valores de libertad y de cultura de masas que demandan el incremento de mayores niveles de comunicación abierta [5].

Como indica el Profesor Castells [5], la aparición de la *sociedad informacional* ha tenido un impacto de una magnitud similar al que tuvo en su momento la Revolución Industrial. La aparición de Internet permitió por primera vez la comunicación de muchos a muchos en un tiempo acordado y a una escala global. Su impacto en la sociedad ha sido tal que actualmente las principales actividades sociales, políticas, económicas y culturales del mundo están presentes y se organizan en Internet. Incluso podemos decir que la economía de los países desarrollados es dependiente de Internet, puesto que la desaparición de esta red ocasionaría una crisis económica inimaginable.

La información y el conocimiento son los ejes que sustentan la nueva sociedad informacional, así como la industria fue lo que sustentó la sociedad industrial. Si el desarrollo de la sociedad giraba en torno a la industria en el siglo XIX y fue la clave para el trabajo y el desarrollo personal de las personas, la adquisición de la capacidad intelectual necesaria para aprender a aprender [6] durante toda la vida, obteniendo información digitalmente almacenada, procesándola y utilizándola para producir conocimientos es un referente obligatorio que introduce la nueva sociedad informacional. Esto constituye uno de los pilares para que se afirme que la re-estructuración fundamental a abordar por la nueva sociedad informacional es la del sistema educativo, mediante la instauración del aprendizaje electrónico (eLearning), que obliga a un uso activo de las tecnologías de la información tanto al profesor como al alumno, facilitando así la introducción de las nuevas capacidades de aprendizaje que las nuevas tecnologías ofrecen.

Como hemos indicado anteriormente, la irrupción de la red Internet en el mundo universitario ha sido espectacular, especialmente desde la aparición de la "Web". La navegación a través de Internet, la sucesiva conexión a múltiples servidores de universidades del mundo, la

obtención al instante de copias de documentos que están a miles de kilómetros y que antes podía llevarnos un año de búsqueda, o el seguimiento de una conferencia en directo desde la otra parte del mundo, hoy es toda una realidad. Las importantes implicaciones que su aplicación supone en el mundo educativo, además de en el mundo laboral, social, familiar...

Pero esto no es lo que entendemos por eLearning. ¿Constituyen las nuevas tecnologías algo más que simplemente un recurso educativo o podrían convertirse en un nuevo medio de enseñanza? Es evidente que el valor del uso de las Nuevas Tecnologías como recurso educativo de apoyo a la docencia es innegable. No sólo por la riqueza de los contenidos multimedia, que facilita la asimilación y comprensión de los contenidos por parte del alumno, sino también por su accesibilidad y disponibilidad. Obviamente, las Nuevas Tecnologías, al igual que los libros de la biblioteca, constituyen un recurso educativo, un complemento de gran valor para la tarea docente, pero además, las nuevas tecnologías nos ofrecen la virtualización de otros recursos presentes en la enseñanza como ciertas tareas administrativas del centro (matriculación, consulta de notas y expediente académico, inscripciones a seminarios, cambios de grupo, petición de certificados, etc.) o incluso en algunas áreas educativas y según qué metodología se emplee, la virtualización del profesor. Con esto, la pregunta aparece rápidamente en nuestras mentes: ¿Podría conseguirse una plataforma tecnológica capaz de sustituir de forma eficaz el papel del centro docente (escuela, facultad, academia...) durante un porcentaje adecuado de la carga lectiva, de forma que no constituya un recurso educativo sino el medio en el que se construye el aprendizaje del alumno?. Este punto de vista es el que vamos a analizar a continuación.

Su consecución parece difícil, al menos en ciertos niveles educativos como la Enseñanza Primaria. En otros niveles, como la Enseñanza Universitaria y cada vez más en Secundaria y Formación Profesional, se usan las Nuevas Tecnologías como soporte al proceso de aprendizaje del alumno, incorporando herramientas como el correo electrónico para consultas, los chats, los foros de debates y otras utilidades de la web junto con unos cursos on-line que proporcionan al alumno la posibilidad de construir su aprendizaje de forma virtualmente guiada. Incluso es posible acceder a una titulación oficial mediante la enseñanza a distancia usando las nuevas tecnologías de la información. Así en algunas de las Universidades españolas como UNED o Universidad Oberta de Cataluña se ofrece esta posibilidad, si bien es obligatoria la presencia del alumno a la hora de realizar los exámenes, prácticas y otros ejercicios de evaluación. ¿Sería posible que Institutos de Enseñanza Secundaria ofrecieran esta posibilidad de educación a distancia o semipresencial para la obtención del título de Bachiller?

Como ya hemos comentado, en otros niveles educativos como el universitario, la formación profesional, la formación post-universitaria o la formación continua, existen ciertas titulaciones o cursos que mediante las nuevas tecnologías permiten su realización a distancia o de forma semipresencial. En la enseñanza Primaria es obvia la imposibilidad de estos procesos debido a la falta de conocimientos previos que el alumno ha de tener para poder hacer uso de las nuevas tecnologías, pero ¿y en la enseñanza secundaria? Teniendo en cuenta la cada vez más temprana y rápida adaptación de los jóvenes del primer mundo a la tecnología y a una sociedad altamente influenciada por ésta, ¿sería posible el uso de una plataforma de enseñanza semipresencial en Secundaria?

1.3.1. Las TIC y la enseñanza secundaria en España

Según el Ministerio de Educación y Ciencia:

“No existe discusión sobre el hecho de que la educación tiene por finalidad la socialización de los educandos, su preparación para la vida como personas responsables, autónomas y libres en un cierto medio social y cultural. En relación con esa finalidad, una sociedad democrática no puede fijarse otro proyecto educativo que el de una educación para la democracia, es decir, para una convivencia ciudadana pacífica, participativa y exenta de discriminaciones”. [7]

Por tanto, desde el estado se está asignando a la institución educativa la socialización de las nuevas generaciones. La escuela no sólo tiene el deber de transmitir conocimientos en determinadas áreas sino también favorecer la incorporación de los individuos a la vida social, creando y fortaleciendo en los individuos las capacidades, habilidades, modos de comportamiento y actitudes necesarias que les permitan participar activamente en la dinámica social en todos sus ámbitos y no sólo en el ámbito laboral. Es por ello que desde el estado, se ha incluido en el Diseño Curricular Base de las Enseñanzas Secundarias asignaturas como "Educación para la convivencia y la paz".

A priori, esta función socializadora podría ser uno de los principales inconvenientes para la creación de un "Instituto Online de Secundaria" puesto el proceso de enseñanza-aprendizaje a distancia eliminaría uno de los principales factores para la socialización de los individuos: la interacción directa, tanto entre iguales como con el conjunto de profesores y demás personal del centro de enseñanza. Y aunque se puede conseguir el contacto indirectamente usando las nuevas tecnologías (chat, audioconferencias, videoconferencias, aplicaciones interactivas, etc.), este no deja de ser un contacto virtual. ¿Podría este modo de interacción influir favorablemente en el proceso socializador de la enseñanza o contribuiría al aumento del fracaso escolar, al aislamiento de determinados alumnos y a casos de inadaptación social, paradójicamente al objetivo que se pretende conseguir? En los últimos años el número de parejas que se conocieron a través de Internet mediante programas de mensajería instantánea, correo electrónico y otros sistemas de telecomunicación, ha aumentado increíblemente, lo que podría suponer un punto a favor de las nuevas tecnologías para el establecimiento de relaciones sociales. Sin embargo en el caso de la educación secundaria, el perfil psicológico de los alumnos, en plena adolescencia y sin una identidad establecida ¿podría influir negativamente? ¿O tal vez sería negativa la falta de uso y adaptación a las nuevas tecnologías lo que provocaría mayor inadaptación social en el futuro tecnológico que les espera?

Según Salinas [8], la llegada de las Tecnologías de la Información y la Comunicación al sector educativo viene enmarcada por una situación de cambios que no pueden ser considerados al margen de los cambios que se desarrollan en la sociedad relacionados con la innovación tecnológica, con los cambios en las relaciones tecnología-sociedad que determinan las relaciones tecnología-educación. La educación no puede ser ajena a estos cambios puesto que los nuevos medios configuran una nueva sociedad [9], a la que el sistema educativo tendrá que servir formando adecuadamente a los alumnos, garantizando su integración en esta nueva sociedad, socializándolos conforme a estos nuevos medios de comunicación, a estas nuevas relaciones sociales, empleando los medios utilizados en la comunicación social.

Otros autores apuntan otras funciones de la institución educativa no reconocidos explícitamente como son la *custodia de los alumnos* [10]. Esta función de guardería, permitió, especialmente en las grandes ciudades, la incorporación de la mujer al mundo laboral; apartar a los jóvenes de los peligros de la calle, y sobretodo disimular el desempleo al retrasar su incorporación al mercado laboral. Decimos permitió puesto que hoy en día la incorporación de la mujer al mundo laboral es casi una obligación tras el incremento del precio de la vivienda, y los peligros de la calle residen a menudo en el camino de casa al centro educativo, en las inmediaciones del mismo o incluso instalados dentro del propio centro. Pretender tener de 8:00h a 14:00h a los hijos adolescentes en el hogar para que por sí solos se preocupen de su formación académica mediante una plataforma de enseñanza a distancia parece ser una utopía, que además puede contribuir a la inadaptación social del alumno.

Otro motivo que dificulta la existencia de una Enseñanza Secundaria Obligatoria a Distancia parte de la dificultad de evaluar ciertos tipos de contenidos. En las propuestas curriculares de la Reforma se distinguen tres tipos de contenidos:

- **Contenidos conceptuales** (hechos, principios y conceptos).
- **Contenidos procedimentales** (habilidades, técnicas y estrategias).
- **Contenidos actitudinales** (actitudes, valores y normas).

Mientras que es viable la transmisión de estos tres tipos de contenidos mediante una plataforma de enseñanza semi-presencial, no lo es tanto su evaluación.

Los hechos se aprenden mediante memorización, tal como son u ocurrieron, y la comprobación es fácilmente automatizable mediante cuestionarios (por ejemplo un test) en los que el alumno tiene que seleccionar la respuesta correcta entre un conjunto de posibilidades dadas.

Los conceptos deben aprenderse de forma significativa. Esto es, el alumno debe comprender lo que está aprendiendo y ser capaz de explicarlo con su propio lenguaje. La evaluación del aprendizaje de los conceptos es fácil si se le solicita que lo explique con sus propias palabras. Esta forma de evaluación es difícilmente automatizable, por lo que requiere la intervención del docente en el proceso de evaluación. De todas formas, también podría automatizarse el proceso de evaluación mediante cuestionarios tipo test, pero su elaboración es ligeramente más compleja que para los hechos pues requeriría ciertas medidas (diferentes cuestiones referidas al mismo principio, cuestiones contradictorias sobre el mismo principio, etc.) para eliminar el factor "suerte" del alumno en las respuestas al azar, y se basaría principalmente en la presentación al alumno de casos en los que aparezcan dichos conceptos.

Los principios son conceptos con un mayor grado de abstracción y que podemos asociar con sinónimos como regla o ley. Suelen describir relaciones de causa-efecto que explican como los cambios producidos en un objeto, un suceso o una situación se relacionan con los producidos en otros. Al igual que los conceptos también deben aprenderse significativamente. La evaluación de los principios sigue la misma pauta que la de los conceptos.

En cambio, los contenidos procedimentales plantean una dificultad añadida no sólo a la hora de su transmisión sino también a la hora de su evaluación. El Diseño Curricular Base define los procedimientos como el "conjunto de acciones ordenadas orientadas a la consecución de una meta". Este tipo de contenidos responde a los conocimientos relativos al *saber cómo hacer las cosas* y hacen referencia a las actuaciones encaminadas a solucionar problemas, para llegar a un determinado objetivo y conseguir de esta forma nuevos aprendizajes. Fuertemente relacionados con los contenidos conceptuales, los contenidos procedimentales pueden ser de componente motriz (manipulación de instrumentos musicales, del instrumental de laboratorio, del ordenador) o cognitiva (habilidades para asimilar conocimientos: realización de esquemas, resúmenes y síntesis, organización de la información, etc.). La principal dificultad a la hora de evaluar este tipo de contenidos, especialmente los que tienen una marcada componente motriz, es que es necesaria la observación directa del docente de la puesta en práctica por parte del alumno. Por ejemplo, en un aula de música, que un alumno sepa correctamente las diferentes técnicas de interpretación en guitarra, con mano apoyada, con mano al vuelo, con púa o sin ella, así como la escala pentatónica, no significa que sea capaz de ejecutar correctamente dicha escala con alguna de las técnicas conocidas. Para evaluarlo es necesario ver como pone en práctica sus conocimientos. Aunque sería posible la observación a distancia, contando con un sistema de videoconferencia que permita al profesor evaluar la interpretación del alumno, los datos que el docente podría obtener para la evaluación del alumno serían menos fiables que si la observación fuera presencial. En estos casos sería recomendable la asistencia al centro de enseñanza del alumno para realizar una prueba de evaluación de contenidos eminentemente procedimentales. Este es un escenario en el que es necesaria la presencia (aunque separados en el espacio) del docente y el discente. ¿Existen otros escenarios posibles en los que el rol del profesor pueda ser asumido por un programa software?

En cuanto a los contenidos actitudinales, su evaluación en un sistema de educación semi-presencial puede interpretarse como el *saber ser y estar* tanto en los actos presenciales como en su presencia virtual (es decir, su comportamiento en la plataforma tecnológica empleada así como su relación con el resto de miembros en dicha plataforma). Nos referimos entonces a los patrones de comportamiento y normas de conducta que permiten al alumno desenvolverse correctamente en el aula virtual. La evaluación de este tipo de contenidos es más compleja, dado que su aprendizaje es lento, los alumnos no siempre manifiestan espontáneamente sus principios o valores éticos. El mejor método de evaluación es la observación sistemática y continua.

Un problema implícito en el uso de sistemas de enseñanza semipresenciales como plataformas para la ESO radica en la observación y apreciación del comportamiento de los alumnos en el entorno virtual y la medida en que éste contribuye a su formación integral como persona, objetivo principal de la enseñanza secundaria.

1.3.2. Infraestructura y acceso a las tecnologías de la información

Aunque además de Internet el Ministerio de Industria, Turismo y Comercio también incluye entre las TIC a otros servicios como telefonía fija, telefonía móvil y televisión digital de pago, vamos a considerar el grado de acceso a ellas por el porcentaje de la población que usa Internet, independientemente del tipo de dispositivo usado para ello.

Según el estudio "Perfil sociodemográfico de los internautas" [11]:

- El 48.3% de los españoles ha usado alguna vez la Red, el 39% usa la Red al menos una vez al mes y un 34% la usa con una frecuencia semanal o menor. Además, va en aumento el número de usuarios pertenecientes a grupos tradicionalmente más alejados de las TIC como la población de edad comprendida entre 45 y 54 años o las personas que se dedican a las labores del hogar.
- En cuanto a la infraestructura y facilidades de contar con acceso a Internet en España, no se aprecia una gran diferencia según el tipo de localidad, manteniéndose en torno a los valores anteriores, observándose un ligero aumento en poblaciones mayores y viceversa.
- El porcentaje de población usuaria de Internet es inversamente proporcional a la edad. Es decir, a mayor edad menor porcentaje de usuarios. Cabe destacar que prácticamente 9 de cada 10 (un 89.2%) jóvenes de edad comprendida entre los 15 y 24 años han usado Internet en alguna ocasión, seis de ellos (un 64%) semanalmente.
- Si comparamos el porcentaje de usuarios según su situación laboral, cabe resaltar que el 64.7% de los activos ocupados han utilizado alguna vez Internet y casi la mitad (46.5%) lo hacen con una frecuencia semanal o menor. Pero sobre todo destaca que el 95.7% de los estudiantes lo han usado, 9 de cada 10 lo usan mensualmente y 8 de ellos semanalmente o a diario.
- También, según el nivel de estudios es destacable que a mayor nivel de estudios mayor porcentaje de población usuaria de la Red: alrededor del 80% de aquellos que tienen una titulación superior universitaria frente a menos del 10% de los que sólo han finalizado los estudios de primaria.

Estos resultados indican un alto grado de implantación y acceso de las TIC en la sociedad española. Especialmente en el contexto de la enseñanza, el grado de acceso y de infraestructura es muy superior (cercano al 100%), lo que nos da idea de la estrecha relación entre TIC y el mundo de la enseñanza. El informe final encargado por la Comisión Europea sobre el acceso y uso de las TIC en las escuelas europeas [3], realizado mediante encuestas a profesores de toda la UE revela que:

- Prácticamente todas las escuelas de la Unión Europea cuentan con equipamiento informático que los alumnos pueden usar, aunque España, con 9.5 ordenadores para cada 100 alumnos, sigue por debajo de la media europea (11.3 ordenadores por cada 100 alumnos).
- La evolución desde el año 2001 al 2006 ha sido muy positiva en España: las escuelas que contaban con una conexión de banda ancha en 2001, un 10%, han pasado a ser un 81% en 2006. Además el número de centros de enseñanza que cuenta con su propia web se ha incrementado en un 10% superando en 2006 la mitad del total de centros de

enseñanza (un 53%). Además en 2006, ocho de cada diez centros de enseñanza cuentan con una red local que los alumnos pueden usar en el centro. A pesar del significativo avance, España sigue estando ligeramente por debajo de la media europea, que lideran los países nórdicos.

- En España, el 85.6% de los profesores usan materiales que han obtenido de Internet para sus clases o apuntes y prácticamente el mismo porcentaje (86.5%) proporciona información a sus alumnos en soporte digital off-line (CD-Rom principalmente). En cambio, el porcentaje de profesores que usa contenidos disponibles en la intranet del centro es menor, un 55.3%.

- En cuanto a la actitud de los profesores con respecto al uso de las TIC por parte de los alumnos, prácticamente todos están a favor del uso de éstas para garantizar la familiarización del alumno con las TIC mediante la realización de ejercicios y prácticas. También un porcentaje cercano al total, afirma que es una buena opción para dejar que el alumno busque información por sí mismo; el 80% de los profesores piensa que las TIC deberían usarse para garantizar un marco de trabajo colaborativo entre los alumnos; y casi el 90% de los profesores afirma que los alumnos se muestran más motivados cuando pueden usar ordenadores en clase.

- Los profesores españoles se destacan con gran diferencia del resto de profesores europeos cuando se les pregunta para qué área está más indicado el uso de ordenadores en la enseñanza. A diferencia del resto de profesores europeos que se divide en diferentes áreas, los profesores españoles se decantan en un 92% por el uso de ordenadores e Internet para enseñar el uso de paquetes ofimáticos (31 puntos por encima de la media europea para esta área). La utilidad de las TIC para la enseñanza de cualquier disciplina es quizás uno de los puntos en los que es más necesario incidir entre los docentes españoles.

- También destaca sobre la media europea (30%) el porcentaje de profesores españoles (40%) que piensa que los contenidos existentes en Internet son de baja calidad (dato que contrasta con el 85% de profesores que usa contenidos de la Red, a pesar de que muchos de ellos piensan que no son de buena calidad).

- Y sobre todo, en este estudio destaca sobre la media europea (un 20% de los profesores) que el 52.3% de los profesores españoles piensan que el uso de ordenadores en clase no tenga beneficios relevantes en el proceso de aprendizaje de los alumnos, dato que contrasta con el 90% de los profesores españoles que piensan que los alumnos se muestran más motivados al usar ordenadores en clase. Teniendo en cuenta que la creación de situaciones de aprendizaje motivadoras es una de las pautas metodológicas fundamentales en el proceso de enseñanza-aprendizaje [12] esta estadística resulta cuanto menos extraña.

2.Estado del arte

2.1.Términos asociados al eLearning

En los últimos años han aparecido sistemas informáticos para la enseñanza y aunque el objetivo de todos ellos es muy similar, los medios mediante los cuales llegan a dicho objetivo varían en gran medida. Muchos de estos sistemas, mal identificados como “*sistemas de eLearning*”, únicamente se centran en la gestión de documentos y en su provisión a discentes y docentes, y aunque ciertamente faciliten la tarea de búsqueda y organización de información, no realizan un seguimiento del proceso de aprendizaje del alumno. Por eso, entender lo qué es y no es el eLearning puede resultar confuso debido a la gran cantidad de términos han aparecido.

Empezamos con la maraña de siglas: muchas personas prefieren la palabra aprendizaje a formación y utilizan el término *technology-based learning* (TBL) o en lugar de *technology-based training* (TBT). Otros términos comúnmente utilizados son *computer-based training* (CBT), *computer-based learning*, (CBL) *computer-based instruction* (CBI), *computer-based education* (CBE), *web-based training* (WBT), *Internet-based training* (IBT), *Hypermedia-based learning* (HBL) o *Multimedia-based learning* (MBL) entre otras. Algunas de ellas pueden verse como subgéneros específicos del eLearning, como por ejemplo WBT.

Otra confusión se debe a las definiciones técnicas que difieren de su uso. Es el caso de CBT, CBI y CBL que se utilizan generalmente para hacer referencia a todos los tipos de eLearning pero se utilizan generalmente para describir la formación basada en discos. Un término que comience con la palabra *computer* por lo general, pero no siempre, se refiere a tutoriales interactivos que se distribuyen en discos. El término formación multimedia se utiliza para describir la formación distribuida a través de CD-Rom.

Por su parte, *browser-based training* (BBT) es un término que se utiliza para describir un curso que requiere un navegador Web para acceder al contenido, pero se puede ejecutar desde un CD-Rom. Estos tipos de cursos se llaman híbridos o CD-Roms híbridos.

Distance learning o *distance education* son otros términos utilizados frecuentemente y aunque describen muchos tipos de eLearning se utilizan para describir generalmente clases a distancia o cursos por correo.

Para complicar un poco más el tema, algunos teóricos dividen el eLearning en tres ramas diferentes: *computer aid instruction* (CAI), *computer-managed instruction* (CMI) y *computer supporter learning resources* (CSLR). El primer término abarca la porción de productos de eLearning que proporcionan enseñanza como tutoriales, simulaciones y ejercicios. El segundo término se refiere a los productos de eLearning que tienen funciones de evaluación, seguimiento y guía de estudio. Finalmente, el tercer término cubre los aspectos del eLearning que dan soporte al desempeño, la comunicación y el almacenamiento. Aunque esta clasificación puede ser útil en el campo de la investigación académica y en foros de discusión, para muchos es suficiente con saber que todas ellas se refieren sólo a partes del conjunto total representado por el eLearning.

2.2.Ventajas del eLearning

El eLearning permite superar algunas de las barreras existentes en los sistemas de enseñanza asistida por ordenador. Algunas de ellas son:

- Elimina las distancias y favorece la movilidad de los usuarios alumnos.
- Aumenta el número de destinatarios que pueden seguir un curso simultáneamente.

- Permite flexibilidad horaria.
- Permite alternar diversos métodos de enseñanza.
- Favorece la interacción entre alumnos. Está demostrado que la no presencia física minimiza la timidez y favorece el establecimiento de comunicación entre los alumnos, especialmente en la adolescencia.
- Anonimato¹
- Seguimiento y tutoría del progreso del alumno a través de los canales de comunicación establecidos.
- Posibilidad de escoger entre gran variedad de materiales, cursos y especialidades.
- Minimiza los costes de formación continua en la empresa.
- Favorece la convivencia familiar para alumnos con responsabilidades familiares a su cargo.

Además de por las ventajas enumeradas, intervienen otros factores que favorecen la implantación de sistemas eLearning:

- Factores económicos: mejor relación coste-beneficio en la producción y desarrollo aprovechando la reutilización de componentes tecnológicos y materiales de aprendizaje. Es un factor interesante a la hora de aumentar los niveles de formación en países en desarrollo, con un alto ritmo de crecimiento económico y con grandes necesidades de trabajadores cualificados.
- Alta disponibilidad de recursos digitales. Las grandes empresas multinacionales necesitan distribuir materiales de aprendizaje a sitios geográficamente dispersos, para que estén disponibles en cualquier momento desde cualquier lugar. La existencia de un gran número de recursos digitales libres y gratuitos en Internet (imágenes, clips de audio y vídeo, animaciones, etc.) favorece su reutilización y aprovechamiento por parte de las grandes empresas (o terceros, como puede ser una empresa especializada en la creación de cursos o implantación de sistemas de eLearning) para la creación de cursos a través de sistemas eLearning.
- Penetración social. La alta penetración en la sociedad de las nuevas tecnologías en general y de Internet en particular, favorece la aceptación de nuevas vías información y de comunicación.
- Ayudas estatales: Los programas de subvenciones por parte del Estado, las Comunidades Autónomas y el Fondo Social Europeo, han incentivado la creación y desarrollo de un sector empresarial dedicado a la formación on-line. Estas subvenciones han hecho posible la aparición de programas como los de Formación Continua de trabajadores, que contribuyen a la adaptación de los trabajadores a las más nuevas tecnologías.

2.3.Inconvenientes del eLearning

No todo son ventajas. Algunos inconvenientes en el empleo de sistemas de eLearning son:

- Preparación del estudiante**: es necesario un esfuerzo para asegurar que los estudiantes tienen las habilidades y conocimientos técnicos, así como el acceso al hardware y software necesarios para completar satisfactoriamente el curso basado en las TICs. Tanto la gestión del tiempo y las habilidades metacognitivas están relacionadas con las actitudes y la motivación del estudiante.
- Personal dedicado**: al igual que los estudiantes, los profesores deben tener habilidades técnicas, conocimiento y acceso al hardware y software, necesarios en este caso, para facilitar el diseño y desarrollo del curso basado en las TICs. Y deben tener un excelente

¹ Como indica Gabriel Kaplun en [1] Capítulo 2, pág24, algunas personas no realizan cursos de formación por vergüenza a reconocer que necesitan seguir formándose, especialmente si son personajes populares.

manejo del tiempo y la motivación para proporcionar asistencia y llevar el seguimiento del estudiante. No obstante algunos autores diferencian rol del profesor, encargado de la selección de contenidos, seguimiento y asistencia al alumno, del rol del técnico encargado del diseño y creación del curso eLearning a partir de los contenidos, objetivos y metodologías, estableciendo de esta forma la necesidad de diferentes perfiles. Un ejemplo de este enfoque puede verse en [13], dónde se describe la organización del equipo de desarrollo e implantación de cursos eLearning de la Universidad de Barcelona:

Equipo Específico	Equipo Común
Coordinador Autor (es) Tutor (es)	Revisores: pedagógico + ortográfico Diseñador gráfico + ilustrador Administrador Tutor (es) Responsable de distribución Secretaría Bedel

Figura 1. Especialización en la creación de cursos eLearning.

Podemos apreciar como se establece una distinción entre el equipo común a todos los cursos, estable, que no depende de un curso concreto y el equipo específico, variable, especializado en la temática del curso en concreto, tratando el desarrollo de cursos eLearning como proyectos software en el que el desarrollo recae en el autor o autores con la ayuda diseñadores gráficos e ilustradores. La práctica docente recae en el tutor o tutores mientras que el coordinador se encarga de la cohesión del grupo completo. Por su parte el equipo estable se encarga de dar soporte a todos los proyectos, incluyendo en este soporte las revisiones ortográficas y sobre aspectos pedagógicos. También se aprecian perfiles específicos para tareas administrativas.

•**Gestión de la información:** a pesar de que se posean unas habilidades técnicas y un manejo del tiempo excepcionales, tanto los profesores como los alumnos requieren de interfaces que reduzcan las cuestiones logísticas y técnicas. El uso de boletines y listas de distribución pueden ayudar a manejar la sobrecarga de información.

•**Equidad:** no todos los usuarios cuentan con las mismas facilidades de acceso a Internet. La tecnología incrementa las diferencias entre los que tienen y los que no tienen tales posibilidades.

•**Ancho de banda:** uno de los mayores inconvenientes hace una década y que está desapareciendo rápidamente con la llegada de líneas de banda ancha. Actualmente, en Europa, el ancho de banda es aceptable y permite transmitir con buenos resultados audio y vídeo sincronizados sin los indeseables "saltos" de antaño.

2.4.¿Por qué usar eLearning?

Las razones para usar eLearning en lugar de educación presencial tradicional son variadas, como varias son las ventajas y los diferentes tipos de eLearning que hemos mencionado anteriormente.

En primer lugar, atendiendo a la primera ventaja enunciada, el uso de la educación a distancia usando nuevas tecnologías minimiza el impacto de las distancias. Y no nos referimos únicamente a la distancia física a la que se encuentra el alumno del centro; la dificultad del alumno para asistir al centro también lo distancia del mismo. Así se ha utilizado eLearning para proporcionar formación en regiones geográficas de difícil acceso, como las regiones húmedas de Guyana o las áreas montañosas de Nepal; o en comunidades dispersas como las islas del archipiélago Indonesio y las pequeñas islas estado del Caribe y del Pacífico. [14]

También se ha utilizado satisfactoriamente en zonas densamente pobladas como China o Pakistán para formar a los profesores siguiendo un programa de formación a gran escala. En otros países se están usando programas de educación a distancia para proporcionar formación a comunidades cuya distancia (no física en este caso) a los centros educativos es difícil de salvar, bien por barreras culturales, políticas, etc. Estos programas juegan un papel muy importante en comunidades marginales como los refugiados en Sudán, las comunidades nómadas de Mongolia y las minorías étnicas del norte de Pakistán.

En cursos cortos de formación continua para trabajadores, la empresa verá reducido los costes directos al usar las nuevas tecnologías para impartir la formación con una plataforma eLearning, y el retorno de la inversión se producirá a corto o medio plazo (más corto cuanto más cursos y más dispersos tanto en tiempo como en espacio).

Otro punto de vista interesante es “que existe la necesidad de adaptar los programas de estudios a las necesidades de las empresas, pero sin reducir la excelencia; por el contrario, dándoles un mayor nivel”[15]

2.5. Especificaciones y estándares en eLearning

Uno de los principales problemas de los sistemas de eLearning siempre ha sido reutilización de los contenidos, de forma que estos puedan ser utilizados en sistemas diferentes, debido a que la mayoría de los sistemas definen sus propios formatos de almacenamiento y procesamiento de los contenidos educativos, así como la forma de acceder y manejarlos. Esta falta de acuerdo se debe en gran medida a la descoordinación en el desarrollo de estándares para eLearning en la década pasada como se verá más adelante.

Hoy en día existen multitud de sistemas destinados a la enseñanza, ya sean meros gestores de contenidos, gestores del proceso de aprendizaje o sistemas más completos capaces de dar soporte a procesos administrativos, ofrecer herramientas de autoría y edición de cursos, etc. Sin embargo, a pesar de la variedad existente, su heterogeneidad dificulta la compatibilidad entre ellos. No todos son de código abierto, algunos usan formatos propietarios y generalmente no es posible re-utilizar contenidos y estructuras de aprendizaje entre ellos.

Estas incompatibilidades, ya sean totales o parciales, repercuten negativamente en el coste asociado a la implantación de un sistema de eLearning, puesto que en el mejor de los casos, una vez superado el tiempo de aprendizaje de las distintas aplicaciones del sistema, sería necesaria la readaptación de material ya existente para otros sistemas, o crear dicho material desde cero en el peor caso.

Una especificación sobre aprendizaje virtual asegura que el nuevo material siga funcionando exactamente igual independientemente de la plataforma que se utilice, siempre que dichas plataformas cumplan la misma especificación.

Los estándares son acuerdos internacionales documentados o normas establecidas por consenso mundial y sólo pueden ser definidos una vez que las especificaciones han sido determinadas, discutidas y validadas, llegándose a un acuerdo para su estandarización por instituciones ampliamente reconocidas como Institute of Electrical and Electronics Engineers (IEEE), Comité Europeo de Normalisation/Information Society Standardization System (CEN/ISSS), British

Standards Institutions (BSI), etc. que en caso de regulación internacional pueden finalmente remitir la propuesta para estándar internacional a la International Organization for Standardization (ISO).

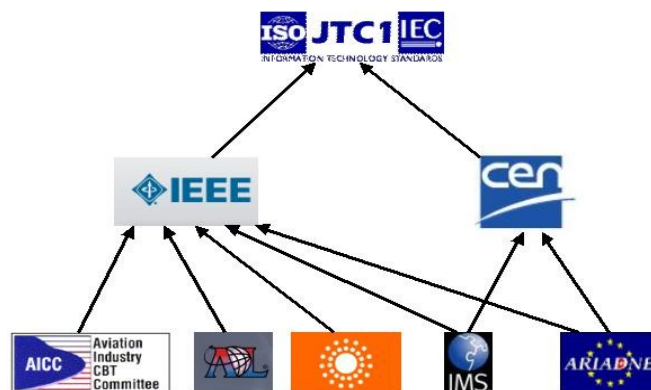


Figura 2. Entidades participantes en estandarización de especificaciones.

El proceso de estandarización es altamente iterativo e involucra a investigadores, usuarios, grupos encargados de la creación de especificaciones y cuerpos de estandarización. El ciclo de vida de la creación de un estándar comienza con una especificación en la que se justifique la necesidad de un estándar. Una vez que esta especificación ha sido ampliamente aceptada por la comunidad, continúa evolucionando y siendo mantenida hasta conseguir un amplio consenso. Este consenso puede producirse rápidamente, por la rápida aceptación de la comunidad (cómo sucedió por ejemplo con el formato de compresión de audio MP3), convirtiéndose en un estándar *de facto*, o lentamente, siendo las organizaciones participantes en el proceso las encargadas del desarrollo de herramientas que sirvan para probar la especificación y servicios que ayuden a crear un mercado entorno a dicha especificación. Una vez validada la especificación por la comunidad y alcanzado el consenso necesario (por ejemplo, para las propuestas de estandarización en IEEE, el 75% de los participantes han de aprobarla), la especificación es propuesta para su estandarización a los organismos (nacionales o internacionales) de estandarización: IEEE, ISO, CEN/ISSS, BSI, etc. Una vez aprobado el estándar, recae en estos últimos organismos la responsabilidad de mantenerlo y actualizarlo a lo largo del tiempo.

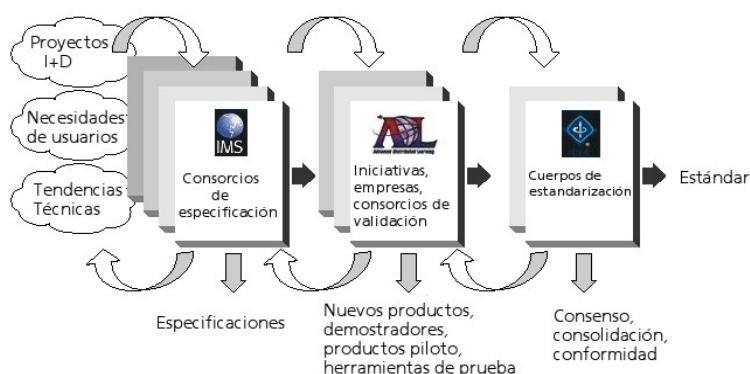


Figura 3. Proceso de estandarización

En la última década y media ha habido muchos desarrollos que han influenciado en la aparición de especificaciones y estándares para eLearning. En este apartado vamos a considerar las principales organizaciones que han contribuido a la creación de especificaciones y estándares en eLearning:

Organización	Descripción y actividades
AICC	Aviation Industries CBT Committee.
IEEE LTSC	IEEE Learning Technology Standards Committee
ADL	Advanced Distributed Learning Initiative.
IMS	IMS Global Learning Consortium
ISO/IEC JTC1 SC36	International Standards Organization
CEN/ISSS WS-LT	Comité Européen de Normalisation.

Tabla 1. Principales organizaciones en la elaboración de especificaciones y estándares en eLearning

Además de las organizaciones de la tabla anterior, ha habido otras cuyo trabajo ha sido fundamental para la industria de las TIC en los últimos años y que han contribuido también al desarrollo de especificaciones y estándares en el campo de la enseñanza:

Organización	Descripción	Web
ARIADNE	Alliance of Remote and Instructional Distributive Networks for Europe	www.ariadne-eu.org
ASTD	American Society for Training and Development	www.astd.org
CETIS	Centre For Educational Technology Interoperability Standards	www.cetis.ac.uk
DCMI	Dublin Core Metadata Initiative	www.dublincore.org
DLF	Digital Library Federation	www.diglib.org
EdNA	Education Network Australia	www.edna.edu.au
OASIS	Organisation For the Advancement of Structured Information Standards	www.oasis-open.org
OKI	Open Knowledge Initiative	www.mit.edu/oki
SIF	Schools Interoperability Framework	www.sifinfo.org
W3C	World Wide Web Consortium	www.w3.org

Tabla 2. Otras organizaciones que ha participado directa o indirectamente en el desarrollo de estándares de eLearning

Algunas de las enumeradas en las tablas anteriores han contribuido directa o indirectamente, alcanzando gran impacto en la industria relacionada con el eLearning. También ha jugado un papel muy importante de forma indirecta World Wide Web Consortium (W3C) involucrado en el desarrollo de las tecnologías de Internet.

Como ya se ha comentado, uno de los principales problemas a resolver era el de la reutilización de contenidos, y esa es la razón de la participación de la mayoría de las organizaciones

en el desarrollo de especificaciones y estándares para eLearning: dar de una u otra forma una solución al problema.

A continuación se muestran las especificaciones más importantes relacionadas con el eLearning que han aportado cada una de las organizaciones citadas. No todas ellas se comentan con el mismo nivel de detalle para no ser repetitivos, puesto que algunas de estas especificaciones han sido definidas a partir de otro conjunto de especificaciones ya existentes, como es el caso de SCORM que utiliza IMS CP, IMS Simple Sequencing, AICC CMI0001, etc.

AICC

La industria de la aviación ha sido tradicionalmente un gran consumidor de formación, por lo que en 1992 decidieron crear un comité que desarrollase una normativa para sus proveedores de formación basada en computador (CBT). De este modo garantizaban la homogeneidad de los requisitos de los cursos y de los resultados obtenidos de los mismos.

Fue el primer organismo destinado a crear un conjunto de normas que permitiese el intercambio de cursos CBT entre diferentes sistemas.

Las especificaciones del AICC² cubren nueve áreas principales, que van desde los objetos de aprendizaje (LO) hasta los Sistemas de Gestión del Aprendizaje (LMS). Normalmente, cuando una compañía dice que cumple con las especificaciones AICC, significa que cumple con al menos una de estas Guías y Recomendaciones (AICC Guidelines and Recommendations, AGRs).

En 1993 AICC publicó la primera versión de CMI Guide For Interoperability (CMI001). CMI estaba relacionado con la gestión tanto de cursos como de alumnos en diferentes sistemas CBT. Las principales funciones de esta guía eran:

- Desarrollo de la estructura de un curso
- Pruebas
- Gestión de usuarios
- Gestión de datos

La terminología usada en las guías AICC CMI está muy influenciada por las expresiones utilizadas en las clases presenciales: lección, currículum, bloque... En las guías CMI se define un conjunto de herramientas para estructurar los cursos de forma jerárquica, identificando la unidad mínima de aprendizaje como la que puede ser seguida y controlada mientras está siendo usada en una lección.

Cada lección del curso tiene una serie de atributos destinados a describirla más detalladamente: objetivos, número de intentos, tipo de lección, reglas de asignación, recursos necesarios, etc. Esto constituye uno de los primeros ejemplos de uso de meta-datos para describir material educativo.

Otros componentes de AICC CMI se ocupan de la administración de alumnos, asignaciones de cursos a éstos, seguimiento del proceso de aprendizaje, procesado de datos de gestión, intercambio de datos entre la lección y el sistema, etc. En la terminología actual, AICC sienta las bases con CMI de lo que hoy denominamos Sistema de Gestión del Aprendizaje o LMS (del inglés Learning Management System).

Por supuesto, la interoperatividad es una de las características de las que se ocupa AICC en sus AGR, las cuales específicamente abordan tres casos en los que una compañía necesita usar lecciones que se crean con diferentes sistemas y herramientas de autor y se ejecutan en un sistema específico:

² Toda la información acerca de AICC ha sido extraída de www.aicc.org

- Migración de un curso de un sistema a otro distinto: La carga sin problemas en un LMS de cursos creados por terceros se consigue definiendo el curso como una entidad totalmente independiente de la plataforma, y creando un sistema (ficheros) de descripción del curso que pueda ser entendido por cualquier plataforma.
- Comunicación entre un sistema y la lección: de tal modo que el curso pueda obtener información necesaria sobre el usuario y después transmitir los resultados de las interacciones y evaluaciones realizadas por el mismo a la plataforma a fin de su almacenamiento y tratamiento estadístico. Para este caso se define un mecanismo de comunicación entre el curso y la plataforma, y un conjunto de datos mínimos que deben ser transmitidos del curso a la plataforma y viceversa. AICC describe dos mecanismos, uno más sencillo y extendido basado en el protocolo http, y otro mediante una API.
- Almacenamiento de los datos de utilización del sistema de un usuario: Los datos referentes a lecciones superadas, lección actual, tiempo invertido, intentos, etc. se almacenan en un sistema de ficheros independiente de la plataforma.

La estructura de un curso determina la forma en que el material será presentado al alumno, en un claro enfoque conductista.

La lista de AGRs es la siguiente:

AGR 001: AICC Publications
AGR 002: Courseware Delivery Stations
AGR 003: Digital Audio
AGR 004: Operating/Windowing System
AGR 005: CBT Peripheral Devices
AGR 006: Computer-Managed Instruction
AGR 007: Courseware Interchange
AGR 008: Digital Video
AGR 009: Icon Standards: User Interface
AGR 010: Web-Based Computer-Managed Instruction
AGR 011: CBT Package Exchange Notification Services (PENS)

AUD001 Plug & Play Guidelines for AICC CBT drivers
AUD002 Digital Audio Portability Guidelines
AUD003 AICC Extensions to the IMA Recommended Practices

CMI001 AICC/CMI Guidelines For Interoperability
CMI003 AICC/CMI Certification Testing Procedures
CMI008 AICC Web-based CMI Certification Testing Procedures

De todas ellas, la AGR010, que habla de la interoperatividad de las plataformas de formación y los cursos, es la que más influencia ha tenido en las especificaciones de eLearning y recientemente la especificación AGR011, una especificación de interoperabilidad que permite que los sistemas de contenidos, como Adobe Captivate³, puedan notificar a los sistemas de gestión de aprendizaje (LMS) que el contenido formativo está listo para su uso e importarlo automáticamente al LMS.

ADL

³ <http://www.adobe.com/es/products/captivate/>

Formada en 1997, la iniciativa ADL⁴ (Advanced Distributed Learning) nació como un programa del Departamento de Defensa de los Estados Unidos y de la Oficina de Ciencia y Tecnología de la Casa Blanca para desarrollar principios y guías de trabajo necesarias para el desarrollo y la implementación eficiente, a gran escala, de formación educativa sobre nuevas tecnologías Web.

Este organismo se apoyó en las anteriores iniciativas para conformar su propio estándar: SCORM, Shareable Content Object Reference Model (Modelo de Referencia para Objetos de Contenidos Intercambiables). Combina muchas especificaciones (de IMS, IEEE y AICC principalmente) y las particulariza para un caso concreto como es el aprendizaje sobre la Web. Las especificaciones, debido a su generalidad, dejan sin concretar aspectos que son necesarios para facilitar la implementación final, y que SCORM trata precisar para lograr una mayor compatibilidad. Concretamente SCORM se apoya en las siguientes especificaciones:

- IEEE Data Model For Content Object Communication
- IEEE Learning Object Metadata (LOM)
- IEEE Extensible Markup Language (XML) Schema Binding for Learning Object Metadata Data Model
- IEEE ECMAScript Application Programming Interface for Content to Runtime Services Communication
- AICC CMI001 Guidelines for Interoperability
- AICC Launch
- IMS Content Packaging
- IMS Simple Sequencing.

Actualmente es el modelo más utilizado en la industria y el que cuenta con mayor cantidad de herramientas que lo soportan.

Esta especificación permite:

- Poder usar un LMS basado en Web para lanzar diferentes contenidos que se han desarrollado por varios autores usando herramientas de autor de diversos vendedores.
- La posibilidad de usar con diversos LMS de diferentes vendedores para lanzar un mismo contenido.
- La disponibilidad de múltiples productos o entornos LMS basados en Web para acceder a un almacén común de contenidos

Con SCORM⁵, ADL propone un modelo de metadatos y estructura de los cursos (CAM), un entorno de ejecución (RTE), y un modelo de secuenciación y navegación de los contenidos (SN), cada uno de ellos organizados en un libro independiente al que se añade un libro con un enfoque más general, que resume toda la especificación (Overview).

⁴ www.adlnet.org

⁵ A partir de la versión 2004, que sustituyó a la versión 1.2 (organizada sólo en dos libros: CAM y RTE) se incorpora la especificación IMS Simple Sequencing, que permite controlar el desarrollo de las actividades de aprendizaje, en un tercer libro.

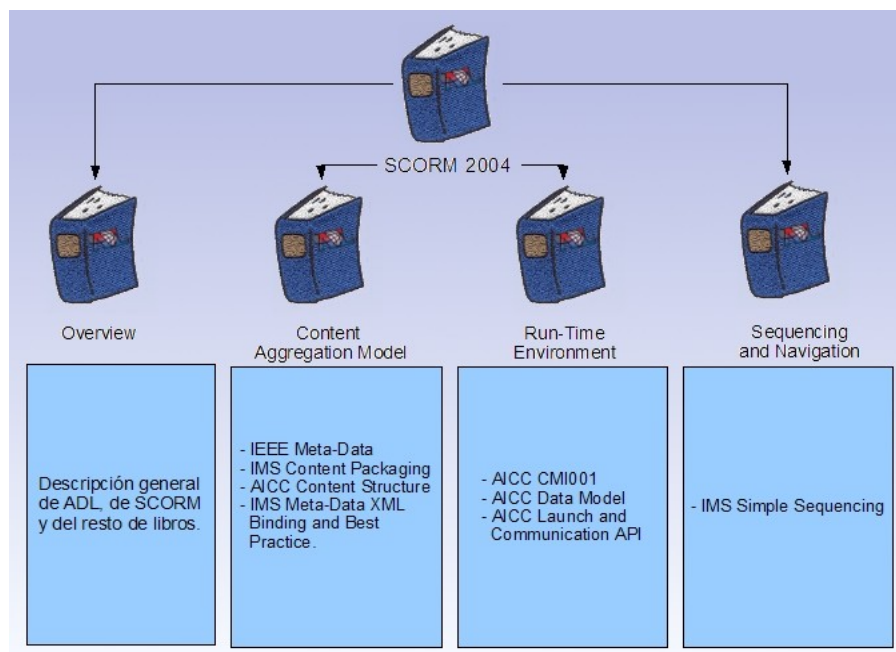


Figura 4. Libros que componen la especificación de SCORM

The SCORM Overview. Contiene una descripción general de la iniciativa de ADL, un análisis de SCORM, y un resumen de las especificaciones técnicas contenidas en las siguientes secciones.

SCORM Content Aggregation Model (CAM). Incluye una guía para identificar y agregar recursos dentro de un contenido de aprendizaje estructurado. En este libro se describe el *SCORM Content Packaging* o empaquetado de contenidos, en el que se identifican los cursos y se distinguen los objetos de aprendizaje compartibles (*Sharable Courseware Object, SCO*), curso o componente de un curso que cumple con los requisitos de interoperabilidad, durabilidad y que dispone de la información suficiente para poder ser reutilizado y accesible. Un SCO es la mínima unidad intercambiable entre sistemas compatibles con SCORM, y consiste en un objeto de aprendizaje que incluye un módulo software (el API de SCORM) que le permite comunicarse con el entorno de ejecución proporcionado por el LMS. Además se identifican los recursos básicos (assets) que son elementos básicos, como ficheros de texto, audio, video, etc. Estos recursos básicos se agrupan en los SCOs. SCORM CP se sustenta sobre la especificación IMS Content Packaging que se detalla en próximas páginas. Técnicamente un SCO es un fichero comprimido (.zip o .jar) que contiene un descriptor conforme a la especificación de IMS Content Packaging.

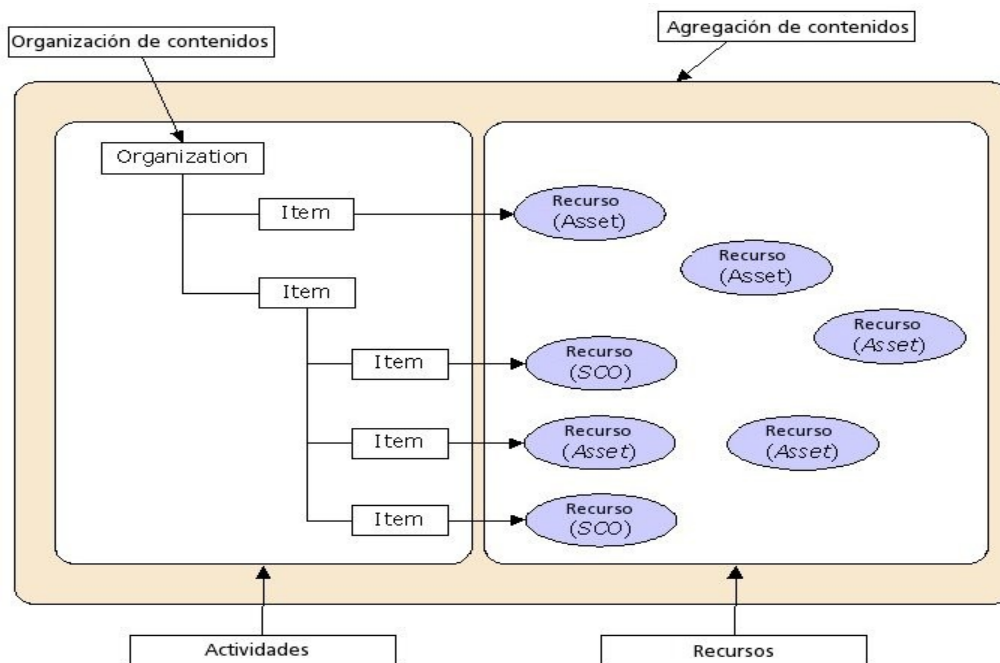


Figura 5. Diagrama ilustrativo de SCORM CAM

SCORM Run-Time Environment (RTE). Este libro incluye una guía para lanzar contenidos y poder realizar un seguimiento dentro de un entorno Web. Toma como punto de partida la recomendación *CM1001 Guidelines for Interoperability* de AICC. RTE proporciona un entorno estándar en el que presentar un objeto de aprendizaje (en este caso un SCO) que es capaz de intercambiar datos con el LMS mediante el API de SCORM (para cuya implementación se usa el lenguaje estándar ECMAScript, más comúnmente conocido por javascript). El LMS se encarga de enviar los contenidos al alumno y el contenido intercambia la información sobre el alumno y el seguimiento de su interacción con el curso al LMS.

SCORM Sequencing and Navigation (SN). Es la información que permite controlar cómo se van a presentar dichos contenidos al usuario (en cuanto a ordenación de los contenidos e interacción con el usuario). Esta presentación no tiene por qué ser siempre la misma, ya que puede depender de las respuestas o comportamiento de los alumnos. Para esta especificación ADL tomé como punto de partida la especificación IMS Simple Sequencing, que veremos a continuación.

IMS

IMS Global Learning Consortium⁶ se formó en 1997 y desde entonces ha sido una de las organizaciones más activas (por no decir la más activa) en el desarrollo de especificaciones y estándares para eLearning, debido principalmente a que está formada por más de cincuenta miembros con intereses directos en la implementación de tecnologías para eLearning (editoriales como Pearson, McGraw-Hill...), desarrolladores de plataformas de eLearning (Angel Learning, Blackboard, WebCT, Sakai...) y otros desarrolladores de software y materiales de eLearning. Aunque algunos miembros compiten en los mismos intereses de negocio el factor de eliminación de la competencia es la colaboración para alcanzar un mayor grado de interoperabilidad, estableciendo una comunidad mayor de usuarios y de productores, intentando imponer sus

⁶ Toda la información acerca de las especificaciones de IMS Global Learning Consortium ha sido extraída de www.imsglobal.org

especificaciones sobre otras comunidades menos potentes, creando así una especie de estándares *de facto* por la fuerza de miembros industriales del consorcio.

IMS elabora sus especificaciones mediante la recolección de requisitos desde sus miembros, usuarios y otros grupos, a través de reuniones que se celebran para atacar aspectos críticos sobre la interoperatividad entre plataformas. Los borradores de las especificaciones, basados en estos requisitos se publican internamente para someterlos a pruebas en las que participan los miembros del consorcio y desarrolladores que colaboran a través de redes de desarrollo establecidas para tal efecto. Una vez que el borrador ha sido probado y validado por el Comité Técnico de IMS se libera como documento público. Una vez publicada la especificación, puede ser propuesta a organismos de estandarización para su adopción como estándar nacional o internacional, si bien lo normal es que transcurra cierto tiempo prudencial, mientras terceras partes que no han participado en el proceso de creación de la especificación hacen uso de ésta, contribuyendo a su futura mejora.

Estructura de las especificaciones de IMS

Normalmente cada una de ellas se encuentra detallada al menos en tres documentos:

Guía de Implementación y consejos. En él se incluyen la forma de uso de la especificación, ejemplos, la relación con otras especificaciones, y cualquier tipo de información complementaria que pueda servir de ayuda. Digamos que se trata de un documento de introducción que servirá para entender los conceptos generales con los que se trata.

Modelo de Información. Documento que describe de manera formal, los datos así como su estructuración, detallando cada uno de los elementos considerados en la especificación. El modelo que se propone en este documento es independiente del formato físico en el que finalmente se representa la información.

Documento de Enlace. Documento que ofrece la forma de representar la estructura de datos de la especificación, generalmente, en XML. Adicionalmente se proporciona el XML Schema que nos permite comprobar la validez de la estructura de un documento que hayamos creado, respecto a la especificación a la que está asociado.

IMS tiene muchas especificaciones ya que cada una de ellas está enfocada en una necesidad distinta del proceso de enseñanza. A continuación vamos a describir con más detalle algunas de las más relevantes.

Meta-Data

La Versión 1.2 fue la primera en hacerse pública (en Junio de 2001). Indica cómo los contenidos deben ser identificados o "etiquetados" y cómo se debe organizar la información de los alumnos de manera que se puedan intercambiar entre los distintos servicios involucrados en un sistema de gestión de aprendizaje (LMS). Tras la publicación del estándar IEEE 1484.12.1 – 2002, IEEE Standard for Learning Object Metadata (LOM) en Julio de 2002, siendo IMS uno de los miembros que contribuyó y participó en su proceso de estandarización, la especificación sobre meta-datos, pasó a denominarse IMS Learning Resource Meta-data y se adaptó al nuevo estándar de IEEE, recibiendo la etiqueta de Versión 1.3.

Content Packaging

Esta especificación provee la funcionalidad para describir y empaquetar contenidos de forma que puedan ser procesados por otro LMS diferente. El empaquetamiento de contenidos está vinculado a la descripción, estructura, y ubicación de los materiales on-line, y a la definición de algunos tipos particulares de contenidos. Esta especificación ha sido comercializada por Microsoft bajo el nombre de LRN (Learning Resource Interchange).

IMS Content Packaging (IMS CP) ofrece una forma de empaquetar en un archivo comprimido tipo .zip (el mismo formato que usan los archivos de distribución de Java o simplemente .jar) los contenidos educativos tales como cursos individuales, conjuntos de cursos, o

cualquier tipo de recurso necesario en el proceso educativo (por ejemplo, evaluaciones o exámenes). Al distribuir una serie de contenidos empaquetados según IMS CP, existe un documento fundamental que es el *Manifiesto*. Dicho documento es un fichero XML, cuyo nombre ha de ser *imsmanifest.xml*, en el que se describe la estructura de los contenidos incluidos en el paquete tal y como podemos apreciar en la figura siguiente.

Meta-data
Organizations
Resources
(sub)Manifest

Figura 6. Estructura del Manifiesto

En el Manifiesto, como veremos a continuación se describen dos niveles diferentes: organización del contenido del paquete y recursos utilizados por dichos contenidos.

Una organización es una posible ordenación jerárquica en forma de árbol de los *Recursos* de un paquete. El estándar permite que un Manifiesto contenga distintas organizaciones sobre los *Recursos* del paquete, dando así lugar a distintas vistas o "cursos" a partir de los mismos contenidos. Esta posibilidad podría facilitar la adaptación del contenido a diferentes dispositivos, diferentes niveles de dificultad, diferentes idiomas, etc. El elemento básico de estructuración que se usa al definir las organizaciones son los *Ítems*. A cada *Ítem* se le puede asociar un *Recurso*, de modo que el árbol de *Ítems* es, efectivamente, una estructuración de los *Recursos* del paquete.

Por su parte la etiqueta *Resources*, engloba la descripción de un conjunto de recursos y sus dependencias. Se puede hacer una relación casi directa entre un *Recurso* y un fichero con contenidos visualizables, como por ejemplo ficheros HTML, animaciones en Flash, imágenes, documentos PDF, etc. En realidad, en cada *Recurso* se puede incluir información sobre los ficheros que lo componen, el tipo de los mismos (que puede ser uno de los tipos ya definidos por el estándar o una extensión de los propuestos) y, opcionalmente, metadatos con información adicional sobre dicho *Recurso*.

En la siguiente página se muestra un esquema que muestra la estructura jerárquica en árbol del Manifiesto.

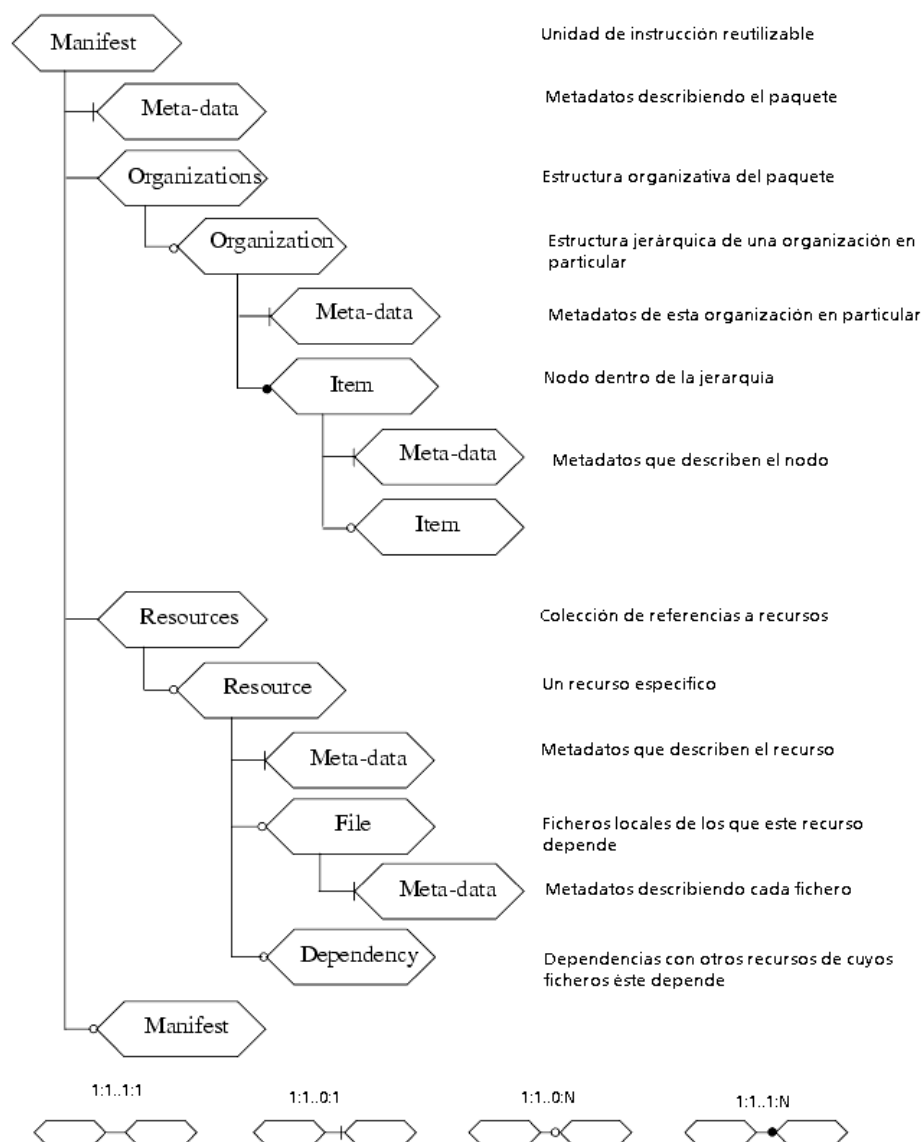


Figura 7. Estructura del fichero imsmanifest.xml

En resumen, el Manifiesto es un fichero XML que describe tanto los contenidos de un paquete como su organización, añadiendo información adicional (metadatos) que pueden ser procesada para llevar a cabo diversas tareas como catalogación de contenidos o selección de un subconjunto de estos según ciertas condiciones (adaptación según el contexto: idioma, edad o nivel de aprendizaje del alumno...). A continuación mostramos un ejemplo de Manifiesto extraído de un curso "demo" creado dentro del proyecto Passepartout⁷ para un curso sobre astronomía y los agujeros negros.

```
<?xml version="1.0" encoding="UTF-8" ?>
<manifest xmlns="http://www.imsproject.org/xsd/imscp_rootv1plp2">
```

⁷ Proyecto perteneciente al programa Eureka-ITEA (código ITEA 04017) de la Unión Europea y financiado por Ministerio de Industria través del programa PROFIT (proyecto FIT-330300-2005-28)

```
xmlns:imsmd="http://www.imsglobal.org/xsd/imsmd_v1p2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
  identifier="MANIFEST-00592D80-D3CF-5079-5EFF-B402BF485260"
  xsi:schemaLocation=
    "http://www.imsproject.org/xsd/imscp_rootv1p1p2 imscp_rootv1p1p2.xsd
    http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2p2.xsd
    http://www.adlnet.org/xsd/adlcp_rootv1p2 adlcp_rootv1p2.xsd">
<metadata>
  <schema>ADL SCORM</schema>
  <schemaversion>1.2</schemaversion>
  <imsmd:lom>
    <imsmd:general>
      <imsmd:title>
        <imsmd:langstring xml:lang="es">
          Astronomía: Agujeros Negros
        </imsmd:langstring>
      </imsmd:title>
      <imsmd:description>
        <imsmd:langstring xml:lang="es">
          Durante este curso, aprenderás aspectos fascinantes de este
          misterioso fenómeno.
        </imsmd:langstring>
      </imsmd:description>
    </imsmd:general>
    <imsmd:educational>
      <imsmd:description>
        <imsmd:langstring xml:lang="es">us_es.curso=1</imsmd:langstring>
      </imsmd:description>
      <imsmd:description>
        <imsmd:langstring xml:lang="es">
          us_es.categoria=astrologia
        </imsmd:langstring>
      </imsmd:description>
    </imsmd:educational>
  </imsmd:lom>
</metadata>

<organizations default="ORG-8BF5F997-218C-409F-2E68-D8BDCDA91707">
<organization identifier="ORG-8BF5F997-218C-409F-2E68-D8BDCDA91707"
  structure="hierarchical">
  <title>Astronomia: Agujeros Negros</title>
  <item identifier="ITEM-FF4E8759-B146-04A6-3688-622C0ABFA9A8"
    isvisible="true"
    identifierref="RES-51D6695D-C43A-89EE-B667-9445C4D17108">
    <title>Astronomía y Agujeros Negros</title>
  </item>
  <metadata>
    <imsmd:lom>
      <imsmd:general>
        <imsmd:title>
          <imsmd:langstring xml:lang="es">
            Astronomia: Agujeros Negros
          </imsmd:langstring>
        </imsmd:title>
        <imsmd:description>
          <imsmd:langstring xml:lang="es">
            Durante este curso, aprenderás aspectos fascinantes de este
            misterioso fenómeno.
          </imsmd:langstring>
        </imsmd:description>
      </imsmd:general>
```

```
</imsmd:lom>
</metadata>
</organization>
</organizations>

<resources>
  <resource identifier="RES-51D6695D-C43A-89EE-B667-9445C4D17108"
type="webcontent"
      adlcp:scormtype="asset" href="astronomia.html">
    <file href="astronomia.html" />
    <file href="astronomia.bmp" />
  </resource>
  <resource identifier="RES-F26C1607-8571-E728-400F-B1422A377CFD"
type="webcontent"
      adlcp:scormtype="asset" href="icon.gif">
    <file href="icon.gif" />
  </resource>
</resources>

</manifest>
```

Listado 1. Ejemplo de Manifiesto IMS CP versión 1.1.2

En la siguiente figura se aprecia el contenido del paquete comprimido y con un formato de intercambio en formato .zip para el curso del ejemplo anterior. En él hemos remarcado el fichero Manifiesto. Los LMS deben por tanto contar con la capacidad de exportar e importar ficheros .zip para poder procesar el contenido en él incluido.

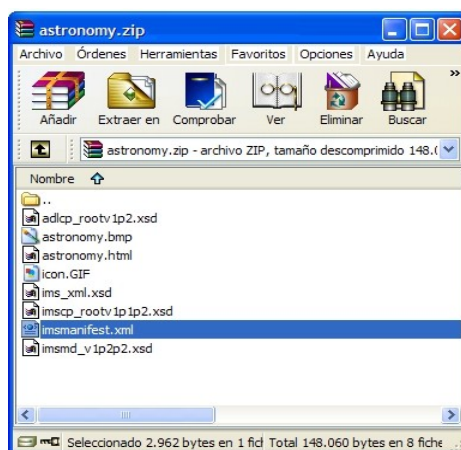


Figura 8. Contenido empaquetado en un fichero .zip

Question & Test Interoperability

Esta especificación describe la forma de representar preguntas individuales o ítems (assessment item) y gestionar evaluaciones o exámenes completos (assessment). Su objetivo es conseguir que tanto las evaluaciones como los resultados sean intercambiables entre los diferentes LMS. Así, podemos disponer de almacenes de preguntas y bases de datos con los resultados obtenidos por los alumnos a los que cualquier sistema de enseñanza electrónica que se habilite podrá acceder.

Con este propósito se plantea y se define una estructura de datos XML para almacenar las preguntas independientemente del sistema o la herramienta de autoría utilizada para crearlas. Esto permite, por ejemplo, la integración en un único LMS de preguntas o exámenes desarrollados con distintas herramientas. Por otro lado se propone un sistema coherente para que los sistemas puedan informar de cuál es el resultado de una evaluación. La primera versión, 1.2, publicada data de 2002 y ha sufrido numerosas actualizaciones y revisiones, hasta la actual versión 2.1 v2 de 2006.

Mientras que en versiones anteriores el principal foco de atención fue cómo presentar la pregunta al usuario, ahora se definen los posibles tipos de interacciones por parte de éste: introducir texto, seleccionar un trozo de texto, seleccionar uno o más elementos de una lista, crear asociaciones entre elementos de dos listas, etc. Además de todas las interacciones contempladas, la especificación introduce un tipo de interacción "abstracta" que los desarrolladores pueden extender y crear nuevas formas de interacción para poder introducir nuevos tipos de preguntas. También tiene plantillas de preguntas para crear preguntas similares, pero en las que hay partes variables que se seleccionan de forma aleatoria entre un conjunto de valores definidos. Otras de las novedades que introduce son los ítems adaptativos, que permiten su corrección adaptativa en función de una secuencia de intentos. Esto permite, por ejemplo, evitar que se le planteen preguntas adicionales al alumno en función de su respuesta actual.

QTI permite la construcción de almacenes digitales de preguntas que sean directamente utilizables en distintos sistemas (podrían incluso utilizarse para generar exámenes tipo test que los alumnos realicen por escrito). El principal problema es que apenas existen almacenes de libre acceso. Un ejemplo de la utilidad de estos almacenes puede verse en la aplicación Comprueba⁸ de la Universidad Complutense cuya finalidad es facilitar el repaso de los contenidos de las asignaturas impartidas en Segundo de Bachillerato LOGSE, de cara a que los alumnos preparen su prueba de acceso a la universidad mediante la realización de exámenes aleatorios generados bajo demanda por la aplicación.

Además de Comprueba no sólo existen LMS comerciales que soportan el formato y la importación de preguntas, sino también LMS de software libre que soportan que además permiten incluso exportar las evaluaciones del sistema en formato QTI (como por ejemplo Claroline⁹ o Moodle¹⁰).

Learner Information Package Specification

Define estructuras en XML para el intercambio de información sobre los alumnos (individuales o grupos) entre diferentes LMS, sistemas de recursos humanos, sistemas de gestión del conocimiento, etc.

La existencia de formatos consensuados para la definición de expedientes de alumnos permite su exportación entre sistemas educativos diferentes. LIP define qué información debe incluirse en el expediente y el formato para representarla. Dentro de los estándares para perfiles y expedientes debe contemplarse tanto la información estática, que no depende de la interacción con el sistema, como pueden ser los datos personales, como la información variable que se genera o se modifica a medida que el alumno avanza en su proceso de aprendizaje, como pueden ser las calificaciones. LIP incluye información de otra especificación sobre información de alumnos denominada *Personal and Private Information* (PAPI) de IEEE y está complementada por otra denominada *Accessibility for LIP* que define nuevas estructuras de datos para poder especificar preferencias de accesibilidad que tengan en cuenta las características del alumno, de modo que el LMS se pueda adaptar a dichas características.

Simple Sequencing

⁸ <http://alamo.sim.ucm.es/comprueba/intro.htm>

⁹ <http://www.claroline.net>

¹⁰ <http://moodle.org>

El objetivo es poder definir el orden en el que se presentan los objetos de aprendizaje o las reglas para seleccionar un objeto de aprendizaje entre varios posibles en función del comportamiento o de las respuestas del alumno. Para ello IMS Simple Sequencing define un modelo sencillo de navegación condicionada, teniendo en cuenta no sólo la estructura del contenido sino además las acciones llevadas a cabo por el usuario. Esta representación de flujo condicionado puede ser creada manualmente o a través de herramientas compatibles con esta especificación. En la especificación se define el comportamiento y funcionalidad que los sistemas han de tener para tal fin. La única versión liberada es la 1.0, que data de Marzo de 2003.

Learning Design

Define cómo describir y codificar las metodologías de aprendizaje y cómo incorporarlas en una solución e-learning. Soporta el uso de un amplio rango de pedagogías para aprendizaje on-line y permite definir nuevas metodologías pedagógicas haciendo uso de un lenguaje genérico y flexible diseñado para permitir la definición de muchas pedagogías diferentes. La Versión 1 es la única hasta la fecha, y fue publicada en Enero de 2003.

IMS Learning Design (IMS LD) ha sido muy bien recibida por los educadores, especialmente pedagogos, pues su objetivo radica más en el diseño de pautas metodológicas que en la mera distribución de los contenidos. Actualmente está recibiendo gran protagonismo y algunos LMS como Moodle comenzaron a soportar esta especificación.

Para entender mejor esta especificación es necesario conocer su proceso de gestación, que comentamos a continuación.

En 1997 la UONL (Open University of the Netherlands) decidió convertir todos sus cursos en cursos on-line. Los cursos existentes empleaban una amplia variedad de enfoques pedagógicos por lo que la UONL decidió clasificarlos y confeccionar unas plantillas representativas de cada enfoque. Aunque el número de plantillas se aproximaba al número de profesores se observó que en la práctica todas consistían en la combinación de tres elementos básicos: recursos educativos, personas actuando con un rol determinado y actividades pedagógicas. Así nació el lenguaje EML (Educational Modelling Language) que permite definir estos tres elementos básicos dando lugar a una Unidad de Aprendizaje o Unit of Learning (UoL).

IMS consciente de las limitaciones pedagógicas de las especificaciones existentes (incluyendo las suyas) trabajó para evolucionar EML. Así nació IMS Learning Design (IMS LD), que entre otras características, soporta procesos de comunicación entre los alumnos, diferencia el rol del profesor, y permite combinar recursos educativos con actividades pedagógicas y con las interacciones entre personas de distintos roles.

Con IMS LD podemos considerar que los recursos son parte esencial para la adquisición de conocimientos por parte del receptor, pero no son suficientes. Es decir, con IMS LD la adquisición de este conocimiento varía en función del uso al que sea sometido el recurso educativo, siendo un proceso más complejo de aprendizaje en el que un mismo recurso educativo puede dar lugar a diferentes conocimientos dependiendo de las actividades realizadas con éstos por profesores y alumnos. Así una unidad de aprendizaje puede tener muchas formas distintas con el mismo recurso. Por ejemplo un texto que describa una ciudad puede formar parte de una unidad de aprendizaje de historia, de geografía, de literatura, de lengua, etc. siendo reutilizable en estas diferentes áreas. IMS LD refuerza la asociación entre *objeto didáctico reutilizable*, *actividades* y *roles de personas* que intervienen en el proceso educativo, asemejándose más al modelo apreciable en las clases presenciales.

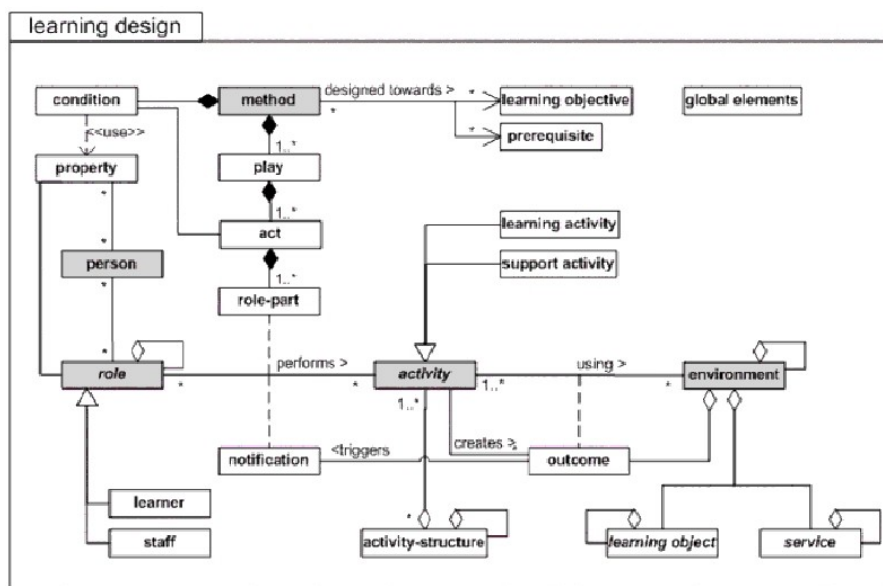


Figura 9. Modelo conceptual de IMS Learning Design

Tal como podemos apreciar en la figura anterior, el diseño de un aprendizaje representa la siguiente idea: todas las personas (person) asumen un rol determinado (role), que puede ser alumno (learner) o miembro del equipo docente (staff). El rol lleva asociado una serie de posibles actividades (activity) que dicha persona puede realizar en busca de la consecución de un objetivo (outcome). Al igual que con los roles, el modelo también especifica una especialización en cuanto a las actividades, que pueden ser de aprendizaje o de soporte. Las actividades se realizan en un entorno (environment) compuesto de servicios (service) y objetos de aprendizaje (learning object) necesarios para que las personas de distintos roles realicen dichas actividades.

Las UoL son más complejas que los objetos de aprendizaje que define SCORM, y permiten oportunidades de aprendizaje igualmente más complejas y ricas. Pensemos en los repositorios de contenidos digitales que, como soporte a la amplia aceptación de SCORM, se han ido creando en los últimos años. Los profesores que usan SCORM a menudo encuentran en estos repositorios la respuesta a ¿qué contenidos didácticos puedo encontrar para enseñar esta asignatura? En cambio, a menudo lo que los profesores más valoran no son los contenidos, sino qué tipo de actividades han realizado otros profesores para impartir la misma materia. Las UoL, debido a la unión en su modelo de estos tres tipos de entidades puede ofrecer soluciones a esta pregunta.

¿Cómo se modela una unidad de aprendizaje según IMS Learning Design?

Básicamente, el principal uso de IMS LD es modelar unidades de aprendizaje introduciendo la información de IMS LD en un paquete de contenidos –preferiblemente, aunque no obligatorio – conforme a la especificación de IMS Content Package (IMS CP), la misma especificación adoptada por SCORM para empaquetar sus SCOs. IMS CP describe sus contenidos por medio de un fichero XML: el Manifiesto, cuyo nombre ha de ser `imsmanifest.xml`, y que se incluirá junto con los recursos necesarios en un fichero comprimido, en formato ZIP (puede ser por tanto un .JAR de Java). El Manifiesto puede incluir “vistas” estructuradas de los recursos contenidos en el paquete. Cada una de estas vistas es descrita como una jerarquía de elementos, a los que se denomina como una “organización”. Cada elemento de la organización hace referencia a un recurso que puede estar incluido en el propio paquete o bien ser un recurso externo accesible mediante una URL.



Figura 10. Estructura de un paquete de contenidos IMS CP

La integración de IMS LD en un paquete de contenidos IMS CP para obtener una Unidad de Aprendizaje, se lleva a cabo incluyendo un nuevo elemento al conjunto de organizaciones: el elemento `learning-design`.



Figura 11. Estructura de una UoL.

Como se aprecia en la figura anterior, una UoL está compuesta por un paquete IMS CP al que se ha añadido un IMS LD dentro del campo Organizations del Manifiesto

El Manifiesto tendría la siguiente estructura:

```
<manifest>
  <metadata/>
  <organizations>
    <learningdesign
      xmlns="http://www.imsglobal.org/xsd/imsl_d_v1p0">
      [aquí se añadirían los elementos de IMS LD]
    </learningdesign>
  </organizations>
  <resources/>
</manifest>
```

Siguiendo la especificación IMS LD, un escenario de aprendizaje, definido por los elementos incluidos entre `<learning-design>` y `</learning-design>`, está compuesto por los recursos, la metodología, las actividades (tanto de aprendizaje como de soporte), la relación existente entre los roles y agrupación de usuarios respecto a dichos roles. El paquete ZIP resultante podrá ser abierto y ejecutado por cualquier sistema compatible con la especificación IMS LD.

La especificación IMS LD define tres niveles o grados conformidad para su implementación:

- Nivel A:** es la implementación básica. Comprende la definición de usuarios, roles, actividades, entornos y métodos. Los usuarios podrán usar diversos recursos externos como enlaces web y algunos servicios como foros y chats para facilitar la colaboración entre estos.
- Nivel B:** Además de todo lo incluido en el Nivel A, añade propiedades y condiciones al nivel A, lo que posibilita la personalización del aprendizaje y formas más complejas de secuenciar las actividades, permitiendo así condicionar el flujo de aprendizaje, esconder y mostrar elementos, incorporar cuestionarios, cálculos numéricos, estadísticas y seguimiento de los usuarios, etc.
- Nivel C:** Añade al Nivel B un mecanismo de notificaciones mediante mensajes, que aporta al sistema la capacidad de iniciar o parar procesos según el cumplimiento de ciertas condiciones, dando soporte así a aprendizajes gestionados por eventos en tiempo-real (por ejemplo aprendizajes interactivos cuyo objetivo es potenciar la destreza del usuario en el manejo de un joystick, entrenadores y simuladores de vuelo, de Fórmula1, etc.)

En los niveles A y B las actividades y el orden de las acciones de los alumnos es predecible, aunque en el nivel B puede condicionarse el flujo de actividades mediante el uso de propiedades y condiciones. En cambio, el nivel C, además permite aprendizajes donde el orden de las acciones y actividades del usuario no es predecible, como en algunos entrenadores y simuladores.

Uno de los principales inconvenientes de IMS LD, es que el enfoque declarativo y el alto nivel de complejidad de la especificación dificultan su implementación. Para reducir errores y facilitar el desarrollo de aplicaciones (editores, reproductores, etc.) conformes con la especificación, se proporciona una descripción de la misma en XML Schema, si bien como se apunta en [21] es difícil describir toda la semántica del texto de la especificación en este lenguaje, y tal vez sería más apropiado usar alguna ontología definida en OWL para ello.

Digital Repositories

La especificación IMS Digital Repositories v1.0, publicada el 30 de Enero de 2003, proporciona recomendaciones para la interoperación entre almacenes de contenidos digitales. Esta especificación define un almacén digital (digital repository) como una colección de recursos que están accesibles en la red sin que fuera necesario a priori el conocimiento de la estructura de dicha colección. Por eso, dicha especificación recomienda su implementación mediante servicios, para garantizar una interfaz común a estos.

Reusable Definition of Competency or Educational Objective Specification

El término competencia aparece como parte de un plan de aprendizaje o de carrera, como pre-requisitos para acceder a un determinado nivel educativo, o como resultados (habilidades, conocimientos, tareas, etc.) obtenidos tras un proceso de aprendizaje. Esta especificación proporciona una nomenclatura estándar para etiquetar los distintos componentes de un sistema de competencias y las características principales de una competencia, independiente de su uso en un contexto en particular, permitiendo así su interoperabilidad entre distintos LMS, sistemas de recursos humanos, etc. IEEE LTSC ha solicitado y obtenido el permiso de IMS para utilizar RDCEO, cuya única versión publicada data de Octubre de 2002, como base para una definición estándar del concepto de *competencia*.

Vocabulary Definition and Exchange

IMS VDEX (Vocabulary Definition and Exchange) define una gramática para el intercambio de listas de valores o vocabularios, que puedan ser procesados automáticamente y entendibles por las personas. Permite por ejemplo definir valores para ser utilizados en IEEE LOM, IMS LIP o en ADL/SCORM.

IMS Enterprise Information Model

Define los modelos de datos que permiten la integración y el intercambio de datos de los LMS con los otros sistemas de gestión de una empresa o centro educativo como, por ejemplo, la gestión de estudiantes o la administración general.

Enterprise Services

IMS Enterprise Services es la definición de cómo los sistemas gestionan el intercambio de información que describen personas, grupos y las adscripciones o pertenencias de las primeras a los últimos en desde el punto de vista organizativo y no educativo, como en el caso de IMS LIP.

ePortfolio

El portafolio electrónico es una colección de documentos en formato electrónico que dan idea de las habilidades, formación y desarrollo profesional de una persona. El concepto en el que se basa es el mismo que cuando se quiere juzgar la calidad de un fotógrafo y se le pide que enseñe sus trabajos previos, o una modelo cuando entrega su *book* a una agencia de modelos. Esta especificación se ha creado para hacer que los portafolios electrónicos se puedan intercambiar entre distintas instituciones y sistemas. El objetivo es lograr que se pueda hacer un mejor seguimiento de las competencias de un alumno, que se mejore su impresión del proceso educativo y su desarrollo personal incluso en formación continua o no reglada. Esto debería simplificar el intercambio de portafolios entre las instituciones educativas y los centros de trabajo. Esta visión del aprendizaje permanente, a lo largo de toda la vida profesional de una persona, se ha denominado *Lifelong Learning*¹¹ y es una prioridad clave para la Comisión de la Unión Europea, que se ha marcado un periodo de 10 años (desde 2000 a 2010) para fomentar la formación continua. Este esfuerzo supone una inversión de 2 millones de euros en 17 proyectos de creación de redes de formación regionales (especialmente en el norte y el sur de Europa), y como objetivo final la creación de un Marco de Calificación Europeo, EQF [26] (European Qualification Framework) que facilite la transferencia y el reconocimiento de las calificaciones obtenidas por los ciudadanos en sus diferentes estudios, reglados o no, a lo largo de su vida.

Shareable State Persistence

Describe una extensión a los entornos de ejecución (como por ejemplo SCORM RTE) que permite el almacenamiento y el acceso compartido a la información de estado entre los objetos de contenido. Trata de solucionar el problema de que un contenido pueda almacenar información de estado en el entorno de ejecución para que pueda ser recuperada posteriormente por ese contenido o por otro. Esta característica es vital para la ejecución de contenido altamente interactivo como, por ejemplo, las simulaciones y hasta ahora se estaba realizando con métodos y formatos propietarios, dificultando la estandarización completa de los sistemas.

Resource List Interoperability

Detalla como intercambiar metadatos estructurados entre sistema que almacenan y proporcionan recursos con el propósito de crear listas de recursos y aquellos sistemas que recogen y organizan estas listas de recursos con un propósito educativo o de entrenamiento. Un ejemplo típico citado en la especificación como lista de recursos es una lista de trabajos o artículos para que lean los estudiantes durante un curso.

AccessForAll Meta-data

Pretende la identificación de recursos que coincidan con las preferencias o necesidades de los usuarios, expresadas usando la especificación IMS Accessibility for LIP. Estas preferencias incluyen la necesidad de utilizar presentaciones alternativas de los recursos, métodos alternativos para

¹¹ Ver informes en: http://ec.europa.eu/education/policies/III/III_en.html

controlar recursos, recursos alternativos a los predeterminados y mejoras o necesidades de ayuda que tenga el usuario. Esta especificación proporciona un lenguaje común para identificar y describir los recursos primarios o por defecto, y las alternativas equivalentes para dicho recurso.

IEEE LTSC

Se trata de un organismo que promueve la creación de una norma ISO, una normativa estándar real de amplia aceptación.

El LTSC¹² se encarga de preparar normas técnicas, prácticas y recomendaciones para el uso del software, herramientas, tecnologías y métodos de diseño que facilitan el desarrollo, despliegue, mantenimiento e interoperatividad.

Está compuesto por más de una docena de grupos de trabajo (working groups o WGs) y grupos de estudio (study groups o SGs) que desarrollan especificaciones para la industria del eLearning.

Los siguientes grupos de trabajo son parte de las actividades generales de la IEEE LTSC:

- IEEE 1484.1 Architecture and Reference Model
- IEEE 1484.3 Glossary
- IEEE 1484.4 Digital Rights Expression Language DREL

Los siguientes grupos de trabajo son parte de las actividades relacionadas con los datos y meta-datos:

- IEEE 1484.12 Learning Object Metadata
- IEEE 1484.14 Semantics and Exchange Bindings
- IEEE 1484.15 Data Interchange Protocols

Los siguientes grupos de trabajo son parte de las actividades relacionadas con los LMS y las aplicaciones:

- IEEE 1484.11 Computer Managed Instruction
- IEEE 1484.18 Platforms and Media Profiles
- IEEE 1484.20 Competency Definitions

LTSC también trabaja en forma coordinada con otra iniciativa denominada ISO JTC1 SC36, que es un subcomité formado conjuntamente por ISO y IEC (International Electrotechnical Commission), dedicado a la normalización en el ámbito de las Tecnologías de la Información para la formación, educación y aprendizaje.

En algunos casos el IEEE LTSC ha comenzado una especificación *desde cero*, pero en otros casos el trabajo ha sido retomado a partir de especificaciones de otras organizaciones como IMS o AICC. Este es el caso particular de los estándares IEEE 1418.11 y P1418.12. En este último, IEEE LTSC creó la noción de meta-dato (información sobre los datos) estableciendo una descripción más detallada de los contenidos del curso, tomando como punto de partida la ofrecida por la AGR010 de la AICC, la cual mejoró.

Learning Object Meta-data

Como ya hemos comentado, los metadatos son datos acerca de los propios datos. Esto es, una forma de definir las características de los datos de forma que nos ayuden en su búsqueda, identificación, clasificación, procesado y representación. Es por tanto, la información relevante para nuestro sistema, extraída de la propia información relevante para el usuario. De forma coloquial, lo

¹² La información sobre estándares de IEEE LTSC puede obtenerse de www.ltsc.ieee.org

que se busca mediante esta información complementaria es poder saber cuál es el contenido y el propósito de un objeto de aprendizaje sin tener que acceder a dicho contenido. Por tanto, los metadatos aportan información orientada a hacer más eficiente la búsqueda y utilización de los recursos. Los metadatos se pueden aplicar tanto a objetos de aprendizaje concretos como a cursos completos o a partes del curso.

Los estándares de metadatos permiten el intercambio de información entre distintos sistemas de e-Learning, independientemente de la implementación de cada uno en cuanto a la forma de almacenar la información.

Uno de los primeros estándares fue el definido por DCMI¹³. Cada uno de los elementos de este estándar está definido por un conjunto de 15 atributos del estándar ISO/IEC 11179, incluyendo por ejemplo: título, nombre, idioma, comentarios o identificador. En su variante, "Simple Dublin Core" se usan únicamente pares atributos-valor. Por ejemplo, para indicar el autor de un documento, se utiliza el atributo `creator` con la siguiente sintaxis: `dc:atributo=valor`. Así, `dc:creator=Márquez`, indicaría el que autor del documento es Márquez. No obstante, el "Qualified Dublin Core" emplea clasificadores adicionales denominados "Dublin Core Qualifiers" (DCQ) para refinar el significado de un recurso.

A partir los esfuerzos previos hechos para la descripción de recursos educativos de DCMI y con la ayuda aportada por IMS y el proyecto europeo ARIADNE, se creó IEEE Learning Object Meta-Data (LOM), actualmente el estándar de e-learning formalmente aprobado que goza de mayor aceptación (estándar IEEE 1484.12.1 – 2002), y que ha sido adoptado en la especificación de IMS Learning Resource Metadata (si bien el proceso de estandarización fue promovido por IMS).

El objetivo de LOM es la creación de descripciones estructuradas de recursos educativos. Su modelo de datos especifica qué aspectos de un objeto de aprendizaje deberían ser descritos y qué vocabularios se pueden utilizar en dicha descripción.

Esta es una descripción jerárquica con nueve apartados principales que agrupan el resto de campos:

- General. Aquí se describe el objeto educativo. Incluye campos como identificador del objeto de aprendizaje, título, descripción, etc.
- Lifecycle. Almacena un histórico del objeto y su estado actual. Detalla quiénes han interactuado con este objeto desde que fue creado, y el tipo de interacción que han realizado.
- Meta-Metadata. Agrupa información sobre los metadatos. Esto puede parecer redundante a primera vista pero resulta muy interesante tener información como quién ha contribuido a la creación de los metadatos y el tipo de contribución que ha realizado.
- Technical. Incluye la información técnica del recurso de aprendizaje, tal como tamaño, ubicación, o formato en el que se encuentra. Además, en este elemento se almacenan los posibles requisitos técnicos necesarios para poder usar el objeto al que se refieren los metadatos.
- Educational. En este elemento se encuentran las diferentes características pedagógicas del objeto. Típicamente se incluyen campos como tipo de recurso –diagrama, ejercicio, figura –, nivel de interactividad entre el usuario y el objeto –alta, media, baja-, o el contexto de uso del recurso – universidad, enseñanza primaria, secundaria, doctorado-, entre otros.

¹³ Dublin Core Metadata Initiative, <http://dublincore.org>

- Rights. Se incluyen los detalles sobre la propiedad intelectual del recurso. También se detallan las condiciones de utilización y el precio en caso de tenerlo.
- Relation. Explica el tipo de relación que tiene el recurso de aprendizaje con otros objetos de aprendizaje. Posee un par nombre-valor en el que detalla el nombre del objeto relacionado y el tipo de relación: es_parte_de, basado_en, etc.
- Annotation. Incluye comentarios sobre la utilización del objeto de aprendizaje, además de su autor y la fecha de creación.
- Classification. Nos informa si el objeto de aprendizaje pertenece a algún tema en concreto. Por ejemplo, es aquí dónde se almacenaría que un objeto de aprendizaje pertenece a la asignatura de Física o a la de Matemáticas, Literatura, etc. Además los temas pueden anidarse, creándose de esta forma sub-categorías como: Historia/Arte_Edad_Media/Pintura/Paises_Bajos.

El modelo de datos indica también qué elementos de la descripción pueden repetirse (cómo acabamos de ver en las anidaciones de Classification) y cuáles no. Además, en algunos campos el contenido es de tipo libre, pudiéndose introducir cualquier cadena de texto (para la cuál se puede especificar además el idioma) y para otros campos se dispone de un conjunto de valores concretos entre los que hay que elegir (es decir, se tiene un vocabulario limitado y controlado por la especificación).

Learning Technology Systems Architecture (IEEE 1484.1:2001)

El estándar 1484.1 del IEEE, LTSA (Learning Technology Systems Architecture), define una arquitectura neutral con respecto a los contenidos, metodologías pedagógicas y la tecnología de la plataforma. Se trata de una arquitectura a muy alto nivel para sistemas de aprendizaje que usen la tecnología como medio principal de transmisión y soporte para los procesos educativos. LTSA proporciona un marco en el que entender los sistemas de eLearning actuales y futuros, pero no especifica detalles concretos para su implementación (como lenguajes de programación, herramientas de autor o sistemas operativos) ni para su gestión (como ciclo de vida de los componentes u objetos educativos, aseguramiento de la calidad, control de acceso o administración de usuarios).

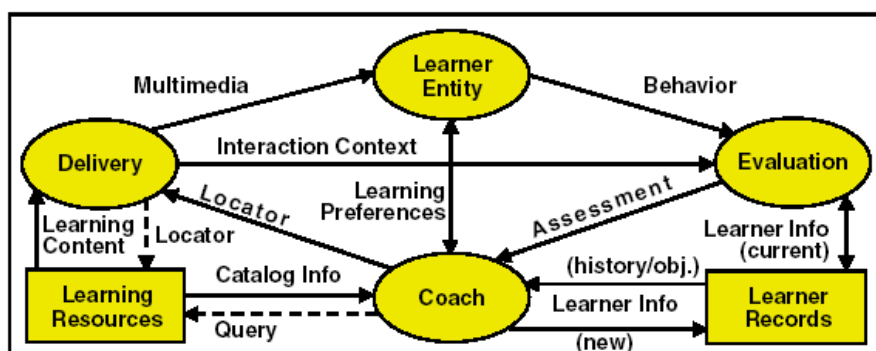


Figura 12. Learning Technology Systems Architecture (IEEE 1484.1)

En el diagrama que define LTSA se diferencian tres tipos de elementos:

- Procesos: Learner Entity, Evaluation, Coach y Delivery.
- Almacenes: Learning Resources, Learner Records

•**Flujos de datos:** Learning Preferences, Learner Info, Assessment, Behavior, Multimedia, Interaction Context, Locator, Catalog Info, Query, Learning Content.

Si observamos el nivel de abstracción con que se describen las entidades de LTSA podremos deducir que no especifica ningún sistema en concreto y que la libertad y ambigüedad que ofrece no facilita la interoperatividad entre diferentes implementaciones de este estándar:

•**Learner Entity:** es una abstracción de un alumno humano. Puede representar un alumno, un grupo de alumnos aprendiendo individualmente, un grupo de alumnos aprendiendo cooperativamente, un grupo de alumnos aprendiendo de una forma heterogénea (algunos colaborando, otros individualmente...), alumnos con distintos roles, etc.

•**Evaluation:** es el proceso encargado de analizar el comportamiento del alumno, teniendo en cuenta el contexto en el que se encuentra y la información relativa al propio alumno.

•**Coach:** negocia las preferencias del alumno y actúa en consecuencia (por ejemplo para mostrar la información en el idioma adecuado o a la resolución preferida por el alumno, etc.). De esta forma se permite establecer el estilo de aprendizaje, que puede ser seleccionado unilateralmente por el alumno (learner entity) o por el sistema (coach), mediante negociación bilateral entre ambos, o bien por una autoridad externa (padres, profesor, institución, desarrollador del curso...)

•**Delivery:** Recupera los contenidos del almacén de contenidos, a partir de su localización y se encarga de su distribución y presentación para el alumno. El mecanismo de distribución puede variar ampliamente al igual que los formatos de presentación, y puede implementarse teniendo en cuenta el proceso Evaluation para conseguir el acoplamiento necesario de cara a obtener experiencias de aprendizaje interactivos.

CEN/ISSS

El Centro Europeo de Normalización, fundado por la Comisión Europea, comenzó su trabajo en 1997.

CEN/ISSS¹⁴ ha establecido un procedimiento para la colaboración y la consecución de consenso denominado CEN Workshops Agreements (CWA), cuyo objetivo es proporcionar a los miembros industriales una herramienta eficiente para establecer acuerdos técnicos donde no es necesario desarrollar un estándar. Parte de este trabajo está relacionado con los requisitos propios de la Unión Europea, teniendo en cuenta la diversidad lingüística y cultural.

La organización ha establecido CWAs en multitud de áreas relacionadas con las TIC, siendo de especial interés el CWA in Learning Technologies. El grupo de trabajo en Tecnologías de Aprendizaje (Learning Technologies Workshop, LTW) se creó al principio de 1999 con el objetivo de potenciar el desarrollo de estándares para eLearning en Europa. Algunos (sería muy extenso listarlos todos) de los CWA publicados por el LTW del CEN/ISSS son:

¹⁴ Toda la información sobre CEN/ISSS así como un ingente número de CWA producidos por este organismo puede consultarse en www.cen.eu

Referencia	CEN/ISSS Workshop Agreements	Fecha
CWA 14590	Descripción de las características del idioma	Octubre, 2002
CWA14643	Internacionalización de IEEE LOM	Octubre, 2002
CWA14644	Estándares de Aseguramiento de la Calidad	Enero, 2003
CWA14645	Disponibilidad de versiones de idioma alternativo en IEEE LOM	Enero, 2003
CWA 15515	Meta-Framework europeo para competencias en TICs. Revisión del estado del arte, aclaración de realidades y recomendaciones para futuros pasos	Febrero, 2006

Tabla 3. Algunos acuerdos producidos por CEN/ISSS

Resumen

Hemos realizado una breve clasificación y resumen de la evolución de las especificaciones según su principal contexto de aplicación, teniendo en cuenta tres contextos generales: contenidos, actividades y metodologías. Pero existen otras áreas o contextos más específicos para los que también han aparecido especificaciones más o menos relacionadas con el eLearning. En la Tabla 1 se muestran las especificaciones existentes clasificadas por área de actuación.

Áreas		Especificaciones
Aspectos culturales, lingüísticos y sociales		--
Accesibilidad		W3C: WAI-AA, WAI-AAA
Competencias		IMS RDCEO
Aspectos pedagógicos		EML, IMS LIP, IMS SS, IMS LD
Perfiles de usuario		IEEE PAPI, HR-XML, vCard
Estructura de los cursos		ADL SCORM, AICC CMI, IMS
Metadatos	Implementación	XML, RDF
	Definición	IEEE LOM, DCMI, CanCore
Empaquetamiento		AICC Course Structure File, IMS CP, IMS QTI, ADL SCORM
Entorno de ejecución		SCORM RTE, AICC RE
Comunicación		HTTP, FTP, SMTP, H.320, H.323, POP 3, IEEE DMCOG, IEEE ECMAScript,
Arquitecturas		IEEE LTSA

Tabla 4. Clasificación de especificaciones de eLearning según área

Es notable destacar que no existen especificaciones ni recomendaciones para homogeneizar la presentación de los cursos independientemente de sus influencias culturales, sociales, religiosas, lingüísticas, etc. Por ejemplo, en muchas culturas orientales, no está permitido representar partes del cuerpo humano separada del cuerpo (forma que adopta el puntero del ratón en algunos sistemas operativos para determinadas operaciones), otras culturas asocian el icono de la calavera y las tibias con espíritus bondadosos y la felicidad en contraposición a la asociación con la muerte, el peligro o algo prohibido en las culturas occidentales. Por eso, a la hora de confeccionar cursos es

necesario tener en cuenta la audiencia a quién va dirigida, para que las metáforas empleadas no lleven a malentendidos o causen un efecto no deseado.

2.5.1. Relación entre especificaciones y procesos de aprendizaje

En el campo que nos interesa (el educativo) la revolución de las nuevas tecnologías y el crecimiento acelerado de Internet han creado la necesidad de que se regulen también sus productos y servicios, por lo que tanto en Europa como, principalmente, en Estados Unidos, se crearon proyectos encaminados a la elaboración de especificaciones y estándares para el diseño de entornos tecnológicos para eLearning.

En el proceso de enseñanza-aprendizaje se pueden distinguir principalmente tres áreas de actuación para el diseñador de un curso:

- Contenidos y recursos:** Engloba a la información proporcionada al alumno. Principalmente formados por páginas HTML, documentos en diferentes formatos (TXT, PDF, RTF, PPT, DOC, SXW...) y recursos (imágenes, archivos de audio y vídeo, enlaces a otras webs, etc.) con los cuales poder crear una presentación multimedia para el alumno. Pueden confeccionarse con diferentes aplicaciones y reutilizarse cuando se necesiten.

- Actividades y funcionalidades:** Las actividades complementan los contenidos y recursos mediante ejercicios y tests que ayudan al alumno a autoevaluarse. También otras funcionalidades como gestión de los procesos de matriculación, del proceso de aprendizaje y expediente del alumno, de la evaluación, etc. también se incluyen en esta área, aunque generalmente estas funcionalidades son ofrecidas por el sistemas de eLearning en el que se desplegará el curso, y si bien no han de ser desarrolladas para cada curso, si que deben ser tenidas en cuenta por los diseñadores del aprendizaje del alumno. En definitiva la funcionalidad ofrecida depende del sistema informático y se trata de un conjunto de servicios que apoyan o complementan tanto a los contenidos y recursos como a la metodología.

- Metodología:** Consiste en el enfoque didáctico. Algunas de las metodologías didácticas más frecuentes en el aprendizaje virtual son: aprendizaje colaborativo, aprendizaje adaptativo, aprendizaje personalizado, etc.

Teniendo en cuenta estas tres áreas, podemos observar la evolución de las especificaciones que han ido apareciendo a lo largo del tiempo, según su área de actuación esté relacionada con contenidos, funcionalidad o metodología. Los primeros trabajos se centraron en aquellos aspectos más concretos como los tipos de archivos que pueden componer un curso, así como la definición y descripción de éstos o su empaquetamiento para poder distribuirlos. Posteriormente la funcionalidad de las plataformas de eLearning para la distribución de los contenidos y la interacción del usuario con el sistema y otros usuarios atrajo la atención de las organizaciones implicadas en la búsqueda de estándares para eLearning. Y hoy en día los esfuerzos se centran en aspectos más abstractos, relativos a aspectos metodológicos: diseño del aprendizaje, diversidad de enfoques pedagógicos, etc. A continuación haremos un repaso cronológico de la evolución de estas especificaciones.

Evolución de especificaciones relacionadas con los contenidos

En la siguiente figura se presenta un cronograma en el que se puede apreciar la colaboración entre distintas iniciativas y proyectos americanos y europeos que resume la evolución de estándares eLearning en el área de contenidos.

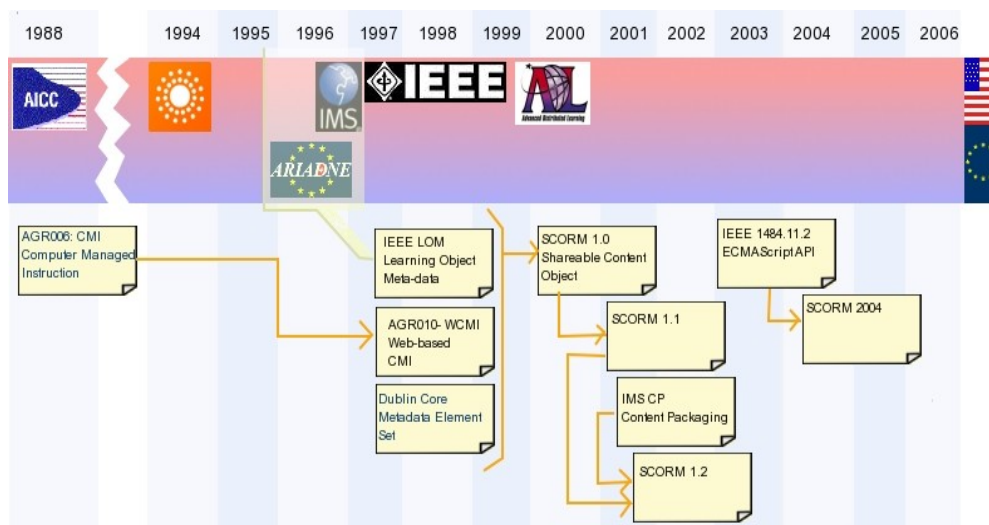


Figura 13. Evolución de especificaciones sobre contenidos en eLearning.

En el cronograma se representan los hitos más importantes en la evolución de las especificaciones para contenidos (definición, descripción, empaquetado, meta-datos, publicación, etc) en eLearning:

En 1988 la AICC (Aviation Industry CBT Committee) se propuso definir una serie de guías y recomendaciones para los cursos de formación a través de ordenadores. Así en 1993, se hizo pública la primera versión de su especificación AGR006: "Computer Managed Instruction (CMI)" que constituía las bases para, con la explosión global de la web, la definición de la especificación AGR010: Web-based CMI en 1998.

Otro organismo que contribuyó al desarrollo y evolución de los estándares en eLearning fue Dublin Core Initiative, creada en 1994 con el objetivo de definir un conjunto de metadatos óptimo para la definición de recursos en la web. La primera versión del "Dublin Core Metadata Element Set" fue publicada en septiembre de 1998. A esta le siguió la versión 1.1 en julio de 1999, que posteriormente ha sido revisada en 2003 y 2004.

Europa y Estados Unidos comenzaron a aunar esfuerzos en la búsqueda de estándares para eLearning. Dos proyectos muy importantes a uno y otro lado del Atlántico son ARIADNE (1996) e IMS (1997). Ambos proyectos colaboraron en 1997 para producir un año después un documento que remitieron al IEEE como propuesta de estandarización, y que contribuyó a la aparición del primer estándar en meta-datos para objetos eLearning: el IEEE LOM. A partir de ahí, se formó la propuesta de estándar o "arquitectura" conocida como la "LTSA" (Learning Technology Systems Architecture) en la que se consideran principalmente tres niveles de análisis: Interacción entre el usuario-alumno y su entorno, Componentes del Sistema (en el ámbito conceptual y de contenidos), y Componentes físicos del Sistema (hardware e interconexión).

Estos documentos citados anteriormente, constituyeron la base en la que se apoyó ADL para definir SCORM (Shareable Content Object Reference Model), que se ha convertido en el estándar de facto en eLearning. En su primera versión de enero de 2000, SCORM 1.0 sólo consistía en una declaración de intenciones y en una ligera especificación del concepto de SCO (Shareable Content Object) como objetos de eLearning que puedan ser compartidos o reutilizados por más de una aplicación o sistema de eLearning. Un año más tarde, ADL publicó SCORM 1.1, en la que se incorporaba un formato en XML para definir la estructura de un curso basado en el de AICC pero

carecía de soporte para metadatos y de un mecanismo robusto de empaquetado de cursos. Esta versión fue reemplazada diez meses más tarde (octubre de 2001) por SCORM 1.2 que se aprovechó de la especificación para empaquetamiento de contenidos publicada en agosto de 2001 por IMS (IMS Content Packaging).

Otro hito importante fue la estandarización de lo que comúnmente se conoce por javascript por parte del IEEE en su especificación "IEEE 1484.11.2 ECMAScript" en el año 2003. ADL había adoptado este estándar como mecanismo de interacción entre el contenido y el sistema gestor de éste, publicando un API para ello e incorporándolo a la especificación de SCORM 1.1. Para la versión de 2004, además de conservar javascript como lenguaje de interacción (aprovechando que se había convertido en un estándar), se definen otros conceptos importantes para este tipo de sistemas, como son la secuenciación, navegación, entorno de ejecución, etc.

Asentado como estándar de facto en el mundo del eLearning, desde su primera versión en enero de 2004, SCORM 2004 ha sufrido ligeras revisiones, dando lugar a SCORM 2004 2nd Edition en julio de ese mismo año y a SCORM 2004 3rd Edition en octubre de 2006.

El respaldo definitivo a SCORM vino de la mano de su principal promotor, el Departamento de Defensa de los Estados Unidos (DoD) que en junio de 2006, mediante la Norma 1322.26 estableció como requisito obligatorio en la instrucción de sus miembros el uso de SCORM en todos los sistemas eLearning.

Como hemos descrito, SCORM es un modelo de referencia para el desarrollo e integración de contenidos de formación para ser utilizados en un entorno virtual de formación. Fue promovido y desarrollado inicialmente por el Departamento de Defensa de los Estados Unidos en 1997, que se aprovechó de "lo mejor" de anteriores iniciativas (el mecanismo de descripción de cursos en XML de IMS, y la API de intercambio de información de la AICC) y creó su propio estándar. Actualmente, SCORM es un modelo aplicable a la formación virtual ampliamente aceptado.

Como puede deducirse de su nombre, SCORM es una especificación centrada en los contenidos. Posteriormente se han añadido nuevas especificaciones para definir la secuenciación de estos contenidos, el entorno de ejecución en el que presentar dichos contenidos y la comunicación entre un sistema gestor del aprendizaje (Learning Management System, *LMS*) mediante un API bien definida, pero todo sigue girando en torno a los contenidos. Aspectos tan importantes como la funcionalidad proporcionada por el LMS o la metodología a seguir durante el curso no forman parte del alcance de SCORM, al menos por el momento.

Evolución de especificaciones relacionadas con la funcionalidad del LMS

Como ya se ha mencionado anteriormente, la funcionalidad de un sistema de eLearning depende principalmente del gestor del aprendizaje elegido. Gran parte de esta funcionalidad puede conseguirse mediante la integración de herramientas existentes, no diseñadas específicamente para eLearning. Khan [16] enumera una lista con algunas herramientas que pueden considerarse como componentes fácilmente integrables en un sistema eLearning:

- Herramientas de comunicación.
 - Asíncronas: Correo electrónico, listas de distribución, grupos de noticias, etc.
 - Síncronas: basadas en texto (chat, IRC, mensajería), videoconferencia, audioconferencia, etc.
- Herramientas de acceso remoto: Telnet, SSH, FTP...
- Navegadores Web.
- Herramientas de búsqueda tanto en el servidor como en la web.
- Aplicaciones cliente-servidor: diccionarios on-line, encuestas, tableros de anuncios y sugerencias, etc.

Cronológicamente, la elaboración de especificaciones relacionadas con el eLearning se centró inicialmente en la localización de recursos de aprendizaje (IMS LOM) y en su descripción y empaquetamiento para facilitar su distribución (IMS Content Packaging). Estos meta-datos incluyen un conjunto de conceptos con valor educativo, tales como edad de los alumnos a los que va dirigido, nivel de dificultad, etc. pero no permiten definir ninguna actividad a desarrollar con los recursos a los que se refieren.

Aunque en algunos casos, a estas actividades se les puede considerar como un contenido o recurso educativo más, existen especificaciones específicas para definirlos. La forma en que se recorren los recursos educativos constituye una actividad muy simple que no podía ser definida mediante las especificaciones anteriores. Este es el objetivo de IMS Simple Sequencing, que permite definir recorridos que el usuario puede seguir a través de los diferentes recursos, permitiendo incluir bifurcaciones según unas condiciones simples. Otra especificación, IMS QTI (Question and Test Interoperability) permite modelar evaluaciones, definir el tipo de problemas, secciones, la presentación y la forma de puntuación de actividades tipo test.

Llegados a este punto, algunos autores [17] advierten de la contradicción entre dos conceptos diferentes de aprendizaje, conductismo y constructivismo, y de la ficticia neutralidad de SCORM frente a esta contradicción de concepciones de la adquisición de conocimiento, en la que se decanta por la primera, sin dar soporte a los pedagogos que quieren implementar recursos educativos reutilizables con una pedagogía más sofisticada (por ejemplo empleando elementos de comunicación y colaboración entre los alumnos). Es por ello que aparece la necesidad de definir nuevas especificaciones que aporten una mayor libertad pedagógica y permitan definir la metodología o metodologías a emplear en el curso.

Evolución de especificaciones relacionadas con la metodología.

Las especificaciones vistas hasta ahora asocian el concepto de "Learning Object" con contenidos, sin ir más lejos, preocupándose únicamente dichas especificaciones de asegurar la interoperatividad y la reutilización de estos "Learning Objects" (LO) entre diferentes plataformas, en un enfoque a priori conductista.

Si analizamos estas especificaciones podemos deducir que interpretan el aprendizaje como un proceso mediante el cual el alumno va "consumiendo" por sí mismo una serie de LO estructurados y secuenciados en un orden determinado. Este engullir de contenidos es continuamente seguido y registrado por el LMS que al final de cada "lote" permite la evaluación de los conocimientos adquiridos por el alumno mediante un test (un LO un poco especial). Debido a este enfoque, algunos autores [18] creen que los LO están simplificando el eLearning, reduciéndolo a un mero seguimiento de páginas, de un contenido "estático, fosilizado y muerto, con una baja motivación y compromiso por parte del alumno, y en un entorno impersonal y aislado".

En realidad, la puesta en práctica del eLearning no gira exclusivamente alrededor de los contenidos, sino que estos son complementados por un conjunto de funcionalidades y servicios como correo electrónico, chats, foros de discusión, etc. sin embargo, los contenidos siguen siendo estáticos y dificultan la "personalización" de estos para adaptarse al alumno en función de su ritmo de aprendizaje (no nos referimos a los intervalos en los que se conecta el alumno para seguir el curso, sino al nivel de dificultad de los contenidos en función de la capacidad de asimilación y aprendizaje del alumno).

Las actuales prácticas pedagógicas no conciben al alumno como una persona aislada, sino que se fomenta el aprendizaje cooperativo, basándose en la razón social del aprendizaje. El modelo constructivista no se basa precisamente en el consumo de contenidos, sino en la propia creación por parte del alumno de su propio proceso de aprendizaje, bien mediante técnicas basadas en la resolución de problemas, síntesis... y por supuesto también, mediante la colaboración entre iguales.

Una pauta metodológica básica en la práctica docente presencial de E.S.O. y F.P. es la personalización y adaptación de los contenidos para alumnos con necesidades especiales o que

muestran un ritmo menor de aprendizaje a la media del resto de alumnos. Sin embargo este comportamiento no es fácil de implementar en un sistema informático centrado únicamente en la distribución de contenidos sin la intervención presencial del profesor.

Para dinamizar el proceso de aprendizaje y facilitar su personalización en tiempo real en función del perfil del alumno, comenzaron a diseñarse especificaciones encargadas de definir las diferentes metodologías aplicables a los procesos de enseñanza a través de las nuevas tecnologías.

Según algunos autores [19], estas especificaciones deben reunir una serie de características:

1. **Complejidad:** debe permitir la descripción completa de un proceso de enseñanza y aprendizaje, sin imponer expresamente una metodología en especial. Dentro de los elementos que debe permitir se encuentran:

- Descripción y asignación de roles de las personas involucradas (alumnos, profesores, tutores, administradores, etc.)
- Descripción e integración de las actividades desarrolladas por el alumno y el cuerpo docente.
- Descripción e integración de los recursos que se usarán al interior del proceso de aprendizaje, siendo los Learning Objects uno de los posibles recursos y totalmente optativos.
- Soporte para distintos modelos de aprendizaje, tanto individual como en grupo.
- Soporte para modelos semipresenciales en los que se complementen procesos de enseñanza a distancia con procesos de enseñanza presencial.

2. **Flexibilidad pedagógica:** La especificación debe además brindar la posibilidad de poder expresar el significado pedagógico de los elementos involucrados en la unidad de aprendizaje, siendo flexible en la descripción de las diferentes clases de pedagogías sin prescribir una aproximación pedagógica concreta.

3. **Personalización:** La especificación debe permitir que los recursos y actividades involucradas en un proceso de aprendizaje se adapten a las preferencias, necesidades, conocimiento previo o circunstancias de los involucrados, sean éstos alumnos, personal docente, o incluso la misma computadora.

4. **Compatibilidad:** La especificación que permita diseñar aprendizaje más complejos deberá integrarse con las especificaciones ya existentes como IMS, SCORM, IEEE.

5. **Reusabilidad e interoperabilidad:** Ha de permitir la reutilización de recursos ya diseñados y creados anteriormente.

De todos los trabajos realizados en esta área, sólo cabe destacar IMS Learning Design, aunque no existen aún muchas herramientas ni LMS que lo soporten, dada la pronunciada curva de aprendizaje y la dificultad de obtener una implementación que respete totalmente la especificación.

2.6. Herramientas

Algunos de los problemas identificados para la amplia adopción de los estándares educativos han sido que algunos LMS han tardado en ser compatibles con las especificaciones, sumado a la necesidad de que los educadores acumulen un alto conocimiento técnico para usar eficientemente el LMS. No obstante, estos problemas están solucionándose poco a poco con la aparición de

nuevas versiones de los LMS, tanto comerciales como de software libre, que soportan el uso de algunas especificaciones y, al menos, la importación o la exportación de cursos completos empaquetados según IMS o SCORM. Además esto está unido también al desarrollo de nuevas herramientas de autor que permiten la creación de objetos de aprendizaje y de cursos completos, y su anotación con metadatos, sin necesidad de ser un experto en los estándares educativos.


En este apartado de herramientas podemos distinguir principalmente entre gestores del proceso de aprendizaje (LMS), gestores de los contenidos de aprendizaje (LCMS) y herramientas de autor. A veces, la diferenciación entre los dos primeros (e incluso para las herramientas de autor) es sólo funcional y en lugar de constituir dos herramientas software diferentes se ofrece en una misma aplicación, que en España se conoce por el nombre de Plataforma Tecnológica. Analizaremos en este apartado las principales soluciones de código libre, que al menos ofrezcan la funcionalidad asociada a un LMS.

Los primeros LMS que aparecieron eran aplicaciones comerciales, que alcanzaron cierta repercusión y reconocimiento en el mundo académico y, en menor medida en el empresarial. Ejemplos de estos LMS fueron WebCT y Blackboard, recientemente fusionados¹⁵. Con los años, el carácter abierto de las especificaciones de eLearning y los procesos de estandarización de algunas de éstas, han contribuido a la aparición de diferentes LMS de código libre, la mayoría de ellos, además, gratuitos.

A la hora de establecer una comparativa de LMS, es necesario tener cuestionar qué puede considerarse un LMS y qué no. Para ello, nos referimos a la recomendación ISO/IEC 9126 [27] que proporciona seis medidas de calidad de cara a la evaluación del software y a la "definición funcional mínima de LMS": funcionalidad, eficiencia, portabilidad, fiabilidad, facilidad de uso y facilidad de mantenimiento. En esta última se hace especial énfasis en las facilidades que aporta el software para modificar su código de cara a adaptarlo a las necesidades del entorno en el que será implantado. En este trabajo de investigación sólo se han tenido en cuenta herramientas Open Source, por lo que en la siguiente relación no aparecen herramientas comerciales como WebCT, Blackboard, CourseGenie, etc. puesto que el autor de este trabajo entiende que este tipo de herramientas comerciales, que no ofrecen libremente el código fuente o no permiten modificarlo con fines no-lucrativos, sirven de freno a la investigación y al intercambio de conocimiento en la comunidad educativa.

Hay muchos proyectos de software libre que proporcionan soporte a los estándares, como por ejemplo, el sistema de ejecución CopperCore para IMS Learning Design desarrollado por los principales creadores de la especificación, la UONL. Además, hay proyectos como SAKAI u Open Knowledge Initiative que tienen herramientas y propuestas de arquitectura para sistemas LMS muy versátiles y en continuo desarrollo. Y además también existen proyectos con una numerosa comunidad de desarrolladores y ampliamente aceptados, como Moodle o .LRN, y proyectos avalados por Administraciones Públicas como OLAT. Sin pretender ser exhaustivos, dada la cantidad de ellos, citamos a continuación los que a nuestro juicio, han tenido mayor repercusión, teniendo en cuenta su denominador común: todos se liberan con alguna licencia Open Source.

¹⁵ <http://www.blackboard.com>

	<p>.LRN¹⁶ Ofrece una completa herramienta para la creación y gestión de portales junto con la capacidad de gestión de cursos, comunidades virtuales, gestión de contenidos y gestión del aprendizaje. Originalmente desarrollado en el MIT (Massachusetts Institute of Technology, USA), hoy en día es utilizado por cerca de medio millón de usuarios en más de dieciocho países. El conjunto de aplicaciones .LRN está respaldado por el Consorcio .LRN cuyos miembros trabajan para dar soporte a las respectivas implementaciones y ampliar la adopción y desarrollo de .LRN. Con este fin, el Consorcio asegura la calidad del software certificando los componentes como <i>.LRN-conforme</i>. .LRN forma parte de OpenACS¹⁷ (<i>Open Architecture Community System</i>), un conjunto de herramientas de software libre para el desarrollo de aplicaciones bajo licencia GPL¹⁸.</p>
---	--


Ficha	
Nombre LMS	.LRN
Soporte	Los usuarios pueden registrarse en http://www.dotlrn.org/register/ y participar en discusiones sobre tele-enseñanza o foros más técnicos.
Última versión	2.2.1
Tecnología utilizada	.LRN está basado en Open ACS (Open Architecture Community System – un conjunto de herramientas para contruir aplicaciones web escalables y orientadas a comunidades) que a su vez se basa en AOLserver.
Sistemas Operativos soportados	OpenACS está diseñado para sistemas tipo Unix. Está desarrollado básicamente en Linux. Puede correr en Mac OS X y en Windows con VMWare.
Requisitos de Bases de Datos/Software	Oracle o PostgreSQL y AOLserver
Tipo de licencia	GPL
Estandares de eLearning soportados	IMS LD (Niveles A y B mediante el paquete IMS-LD 1.1.0) SCORM 1.2
Idiomas	Inglés
LCMS	Sí. Permite importar cursos SCORM 1.2
Extras	<ul style="list-style-type: none"> - Weblogs personales - Foros - Chats - E-mail - Agenda/Calendario - Encuestas - Álbum de fotos - Tablón de Noticias - Almacén de contenidos - Zona de almacenamiento de ficheros - Tests y evaluaciones - Motor de búsqueda
Notas	<p>Usado por UNED¹⁹ y Universitat de València.</p> <p>Proceso de instalación excesivamente complejo.</p>

¹⁶ <http://dotlrn.org>

¹⁷ <http://openacs.org>

¹⁸ <http://www.gnu.org/licenses/licenses.es.html>

¹⁹ Universidad Nacional de Educación a Distancia. <http://www.uned.es>

	<p><u>ATutor</u>²⁰</p> <p>Este proyecto empezó en 2002 en colaboración con el Adaptive Technology Resource Centre (ATRC) de la Toronto University. Este centro es un líder internacionalmente reconocido en el desarrollo de tecnologías y estándares que permitan a la gente con discapacidades el acceso a las oportunidades elearning y esta misión ha influenciado profundamente el desarrollo de la plataforma. El desarrollo ha prestado especial interés a la accesibilidad: ATutor es la única plataforma LMS que cumple las especificaciones de accesibilidad W3C WCAG 1.0 de nivel AA+.</p>
---	--


Ficha	
Nombre LMS	ATutor
Soporte	http://www.atutor.ca/contact.php?subject=ATutor+Service+Request
Última versión	1.5.3.3
Tecnología utilizada	PHP, Apache, MySQL
Sistemas Operativos soportados	Todos los sistemas operativos en los cuales es posible instalar PHP, Apache y MySQL (ej. Windows, Linux, UNix, etc)
Requisitos de Bases de Datos/Software	Apache 1.2.x, PHP 4.2 o superior (con soporte habilitado para zlib y mysql) y MySQL 3.23.x o superior.
Tipo de licencia	GPL
Estandares de eLearning soportados	SCORM 1.2 LMS RTE3 W3C WCAG 1.0 de nivel AA+ W3C XHTML 1.0
Idiomas	>15 Alemán, Catalán, Danés, Inglés, Holandés, Francés, Griego, Italiano, Portugués, Ruso, Español, Koreano, Letón, Noruego, Tailandés, Turko.
LCMS	Sí. Se pueden gestionar cursos, es fácil crear contenidos y tests dentro de la plataforma e importar paquetes SCORM 1.2.
Extras	<p>Permite desarrollar módulos instalables a modo de plugins en el servidor:</p> <ul style="list-style-type: none"> - Control de pagos (para cursos de pago). - Acceso a almacenes de contenidos: MERLOT, The Learning Edge (TLE) - CMAP (Herramienta colaborativa para crear mapas conceptuales) - Álbum de fotos - Vídeo conferencia. Existen varios módulos: <ul style="list-style-type: none"> - Marratech²¹) - Elluminate Live!²² - Text To Speech: Atalker - Web chat. - Webcast - Calendario - Weblogs personales: Ewiki²³ - Temas (los usuarios pueden personalizar la apariencia y guardar la configuración) - Motor de búsqueda - Tests y evaluaciones
Notas	

²⁰ <http://www.atutor.ca>

²¹ <http://www.marratech.com>

²² <http://www.illuminate.com>

²³ <http://erfurtwiki.sourceforge.net/>

	<p><u>Claroline</u>²⁴</p> <p>Claroline es uno de los LMS más usados en el mundo. Muchas universidades aprecian su entorno de aprendizaje colaborativo que permite a los docentes y a las instituciones educativas crear y administrar cursos en la web. El sistema ofrece, entre otras, herramientas de gestión de los grupos, forum, almacenes de documentos, chat y administración del perfil de los usuarios.</p>
---	---


Ficha	
Nombre LMS	Claroline
Soporte	info@claroline.net
Última versión	1.7.0
Tecnología utilizada	PHP, Apache, MySQL
Sistemas Operativos soportados	Todos los sistemas operativos en los cuales es posible instalar PHP, Apache y MySQL (ej. Windows, Linux, UNix, etc)
Requisitos de Bases de Datos/Software	Apache, PHP 4.1.0 or posterior (con soporte habilitado para zlib y mysql) y MySQL 3.23.6 o posterior
Tipo de licencia	GPL
Estandares de eLearning soportados	SCORM 1.2
Idiomas	> 30
LCMS	Se permite gestionar cursos.
Extras	<ul style="list-style-type: none"> - Web chat - Foros - Wiki - Importación de cursos SCORM - Tests y evaluaciones
Notas	No conforme con SCORM RTE3, no personalizable por los usuarios, no hay motor de búsqueda.

²⁴ <http://www.claroline.com>

CopperCore	<p><u>CopperCore</u>²⁵</p> <p>Es un motor de la especificación IMS LD capaz de ejecutar Unidades de aprendizaje de los tres niveles (A, B y C). Esconde la complejidad de la ejecución para que sea utilizada por los desarrolladores. Utiliza Java y XML como base para programación e intercambio de información entre entidades, ya que IMS LD está basado en XML. Utiliza propiedades (variables) como mecanismo de almacenamiento de datos</p>
-------------------	--

Ficha	
Nombre LMS	CopperCore
Soporte	En la web: http://coppercore.sourceforge.net/
Última versión	3.0
Tecnología utilizada	J2EE, Base de datos relaciona (Probado con MS SQL Server/MSDE, PostgreSQL y HSQLDB). Diseñado para ser instalado sobre Jboss 3.2.x, pero puede usar cualquier otro servidor de aplicaciones.
Sistemas Operativos soportados	Cualquiera para el que exista una máquina virtual de Java.
Requisitos de Bases de Datos/Software	Aunque puede funcionar sobre otro software compatible, esta son los garantizados: Java 1.4.x Microsoft SQL Server 2000/MSDE 2000 + driver JDBC PostgreSQL 7.3 + driver JDBC HSQLDB 1.7.2 Apache Derby (también conocido como IBM Cloudscape) El driver JDBC para HSQLDB y Derbi se distribuye con ellos. Apache Axis 1.1 SOAP toolkit for Java. Jboss 3.2.6
Tipo de licencia	GPL
Estandares de eLearning soportados	IMS LD (Niveles A, B y C) IMS ePortfolio IMS QTI v 2
Idiomas	Inglés
LCMS	No.
Extras	<ul style="list-style-type: none"> - CopperCore Kernel: Núcleo de CopperCore que contiene toda la funcionalidad necesaria para procesar IMS LD en cualquiera de sus tres niveles. - Publisher: Un servlet que proporciona una primitiva interfaz web para la validación y publicación de paquetes IMS LD. - CLICC: Interfaz de línea de comandos que permite la ejecución de varias tareas administrativas. - WebPlayer: Interfaz web para el entorno de distribución de IMS LD. Actualmente sólo funciona en Microsoft Internet Explorer. - CopperCoreSOAP: Una capa conforme con la especificación de W3C SOAP. Permite a aplicaciones externas comunicar con el núcleo de CopperCore a través de SOAP sobre HTTP.
Notas	Colabora con Reload editor para la consecución de un editor de IMS LD


²⁵ <http://coppercore.sourceforge.net/>

	<p>Dokeos²⁶ El proyecto empezó desde una versión previa de Claroline y ha conseguido tener entidad por sí mismo. El objetivo es ayudar al docente a crear contenido pedagógico, a estructurar las actividades en caminos de aprendizaje, a interactuar con los estudiantes y a seguir su evolución mediante un sistema de informes.</p>
---	---

Ficha	
Nombre LMS	Dokeos
Soporte	info@dokeos.com
Última versión	1.8.1 RC1
Tecnología utilizada	PHP, Apache, MySQL, Adobe Flash
Sistemas Operativos soportados	Todos los sistemas operativos en los cuales es posible instalar PHP, Apache y MySQL (ej. Windows, Linux, UNIX, etc)
Requisitos de Bases de Datos/Software	Apache, PHP 4.x o posterior (con soporte habilitado para mysql, zlib, preg y xml) y MySQL 3.23.6 o posterior.
Tipo de licencia	GPL
Estandares de eLearning soportados	SCORM 1.2
Idiomas	> 30
LCMS	Se permite gestionar y crear cursos, así como crear páginas (CMS) y tests. Permite importar y exportar cursos SCORM 1.2
Extras	<ul style="list-style-type: none"> - Web chat - Foros - Vídeo conferencia (basada en flash) - Tests y evaluaciones en línea - Herramienta de autor: Oogie (Convertor de PowerPoint y OpenOffice Impression) - Encuestas - Motor de búsqueda - Permite definir itinerarios de aprendizaje (Learning Paths)
Notas	Dokeos ha sido seleccionada como plataforma de gestión de cursos para el proyecto europeo e-Hospital ²⁷ , que pretende evaluar la viabilidad del uso del e-learning en la formación de pacientes adultos hospitalizados.


²⁶ <http://www.dokeos.com>

²⁷ <http://www.ehospital-project.net/>

	<p>Eledge²⁸ Este software, desarrollado por la Universidad de Utah, es un entorno de creación de un websites para la educación en línea que incluye el registro de estudiantes, la autenticación, la creación de contenido, encuestas, exámenes, evaluación de trabajos, introducción de informes, libro de notas del instructor, calendario de clase y ayuda en línea.</p>
---	---


Ficha	
Nombre LMS	Eledge
Soporte	wight@chem.utah.edu .
Última versión	3.1.0
Tecnología utilizada	Java Servlets, Apache, MySQL
Sistemas Operativos soportados	Todos los sistemas operativos en los cuales es posible instalar un contenedor de servlets, Apache y MySQL (ej. Windows, Linux, UNix, etc)
Requisitos de Bases de Datos/Software	Compilador/intérprete Java, servlet container (Tomcat ,Jetty, JonAs, JBoss, etc), MySQL, Apache webserver
Tipo de licencia	GPL
Estandares de eLearning soportados	Ninguno
Idiomas	2 (Inglés y Español)
LCMS	No
Extras	<ul style="list-style-type: none"> - Herramienta de creación de tests. - Permite subir contenidos (HTML) - Encuestas - Evaluación de trabajos - Calendario
Notas	No conforme con los estándares. Ni siquiera puede importar SCORM.

²⁸ <http://eledge.sourceforge.net>

	<p>ilias²⁹</p> <p>Es un producto creado en la Universidad de Colonia a partir del proyecto VIRTUS. Permite al usuario crear, editar y publicar unidades de cursos integrados con sus navegadores. Se trata de una plataforma que ofrece muchas funcionalidades a todos los niveles. Con Ilias es posible establecer diferentes escenarios e incluso entornos complejos para todos los usuarios.</p>
---	---


Ficha	
Nombre LMS	Eledge
Soporte	ilias-info@uni-koeln.de
Última versión	3.7.6
Tecnología utilizada	PHP, Apache, MySQL
Sistemas Operativos soportados	Todos los sistemas operativos en los cuales sea posible instalar PHP, Apache y MySQL
Requisitos de Bases de Datos/Software	Apache 1.3.29, MySQL 4.0.18, PHP 4.3.6, y algunas librerías: libgd, libz, libpng, libjpeg, ... (la lista completa está en la web)
Tipo de licencia	GPL
Estandares de eLearning soportados	SCORM 1.2
Idiomas	>15 Albanés, Alemán, Búlgaro, Chino, Checo, Danés, Inglés, Español, Estonio, Francés, Griego, Húngaro, Italiano, Polaco, Portugués, Rumano, Ruso, Serbio.
LCMS	Sí, se pueden gestionar cursos y soporta múltiples clientes que administren múltiples bases de datos. La plataforma Ilias también permite crear contenidos.
Extras	<ul style="list-style-type: none"> - Entorno integrado de autor para crear cursos. - Soporte de metadatos para todo tipo de 'learning objects' - Múltiples clientes soportados en la misma instalación - Muchas funcionalidades para usuarios y administradores - Funcionalidades de 'Role Based Access Control' - Módulos de importación SCORM/AICC/HTML - Herramientas de test y evaluación disponibles
Notas	

²⁹ <http://www.ilias.de>

	<p>Moodle³⁰ Moodle es el acrónimo de "Modular Object-Oriented Dynamic Learning Environment". Se trata de un software libre para la producción y realización de cursos en línea, además de páginas web. Es uno de los LMS más populares y está actualmente viviendo una fase explosiva de expansión. Su comunidad de usuarios y desarrolladores es muy numerosa y se caracteriza por su entusiasmo respecto al sistema. Moodle es un proyecto inspirado en la pedagogía del constructivismo social.</p>
---	---

Ficha	
Nombre LMS	Moodle
Soporte	A través de los Moodle Partners (empresas de servicios)
Última versión	1.7.1
Tecnología utilizada	PHP, Apache, MySQL
Sistemas Operativos soportados	Todos los sistemas operativos en los cuales sea posible instalar PHP, Apache y MySQL
Requisitos de Bases de Datos/Software	Apache 1.3.29 o Apache 2, MySQL 4.1.16, PHP 4.x o PHP 5 (este último sólo a partir de la versión 1.4 de Moodle), y algunas librerías: libgd, freetype, mbstring, zlib, phpldap, etc
Tipo de licencia	GPL. Certificación OSI (Open Source Initiative)
Estandares de eLearning soportados	SCORM 1.2 IMS-QTI
Idiomas	> 40
LCMS	Permite crear páginas webs e importar cursos SCORM, pero no permite exportarlos.
Extras	<ul style="list-style-type: none"> - Instalación de temas para personalización - Multitud de Plugins: (Ver http://moodle.org/mod/data/view.php?id=13&page=0) <ul style="list-style-type: none"> •Google Translator •AIM Messenger •Grabador de audio en MP3 •Generador de Certificados en PDF para los alumnos •Chat •CMS (para crear páginas web) •Wiki •Conexión a almacenes de contenidos (Learning Object Repositories) •Video conferencia: illuminate •Foro •Podcasts •y muchos más.
Notas	Ha tenido una amplia aceptación y se ha extendido en multitud de centros educativos. Ver: http://www.moodle.org/sites


³⁰ <http://moodle.org>

	<p>OLAT³¹ Olat es un LMS/LCMS muy completo desarrollado por la Universidad de Zurich y utilizado por la Administración Pública suiza. Está desarrollado íntegramente en Java y se distribuye junto con su código fuente sin coste alguno. Una de sus características es la calidad de su diseño, que permite varios mecanismos para extender la funcionalidad del sistema con nuevos módulos. También permite la instalación de la plataforma en "modo desarrollo" que permite presentar en la propia web, información detallada de los eventos que el sistema atiende y su respuesta. No utiliza ninguna tecnología para el cliente: No usa applets, ni flash, ni ningún otro plugin que el usuario deba instalar en su máquina (en el lado del cliente, claro).</p>
---	--

Ficha	
Nombre LMS	OLAT
Soporte	Soporte oficial: Departamento Multimedia & e-Learning Services de la Universidad de Zurich. http://www.id.unizh.ch/dl/elearning.html Soporte comercial: Frentix GmbH
Última versión	5.1
Tecnología utilizada	Java y alguna base de datos relacional (garantizado con MySQL, PostgreSQL y HSQLDB, aunque también funciona con ORACLE)
Sistemas Operativos soportados	Todos para los que exista máquina virtual de Java.
Requisitos de Bases de Datos/Software	Java 1.5, MySQL 5
Tipo de licencia	GPL. Certificación OSI (Open Source Initiative)
Estandares de eLearning soportados	SCORM 1.2 RTE3 IMS QTI v1.2.1 iCalendar (RFC 2445)
Idiomas	8 Alemán, checo, español, francés, griego, italiano, inglés, polaco.
LCMS	Sí.
Extras	<ul style="list-style-type: none"> - Seguridad basada en permisos (más flexible que por roles) - Sistema virtual de ficheros (WebDAV) - Portal personalizable para cada usuario - Espacio para compartir ficheros para cada usuario - Motor de búsqueda para diferentes formatos de fichero - Calendario (iCal) - Chat - Wiki - Almacén digital de recursos para compartir con otros autores - Catalogación de recursos - Mensajería Instantánea (Jabber³²)
Notas	Protocolo de autenticación y autorización Shibboleth totalmente integrado. Eficiente, ligero y con una interfaz gráfica muy atractiva.

³¹ <http://www.olat.org>

³² <http://www.jabber.org/>

	<p><u>Sakai</u>³³ Producto elaborado conjuntamente por las Universidades de Michigan, la Universidad de Indiana, el MIT, el uPortal Consortium y la Open Knowledge Initiative (OKI). Estas instituciones se unen para sincronizar sus fuerzas y elaborar una colección pre-integrada y modular de herramientas de código abierto.</p> <p>El proyecto Sakai se desarrolla para cubrir específicamente las necesidades de la educación superior. Programado en Java, y sobre Tomcat, es el LMS de OKI. A través de un portal académico, permite la gestión de cursos con herramientas de evaluación, proporciona un sistema colaborativo de apoyo a la investigación y un motor de flujos de trabajo (<i>workflows</i>). Se basa en la arquitectura Technology Portability Profile (TPP) y está creado a partir de un conjunto de herramientas de software educativo que integra las mejores características de los distintos softwares de sus socios a fin de crear un portal institucional basado en servicios desde el que acceder a las distintas herramientas.</p>
---	---

Ficha	
Nombre LMS	Sakai
Soporte	www.okiproject.org
Última versión	2.3.1
Tecnología utilizada	Java
Sistemas Operativos soportados	Cualquiera para el que exista una máquina virtual de Java.
Requisitos de Bases de Datos/Software	Java Run-Time Environment (JRE) o Java Software Development Kit (J2SDK) versión 1.4.2 o posterior. Apache webserver y Tomcat 5.0.25 o posterior. Base de datos MySQL (version 4.1. o posterior) u Oracle
Tipo de licencia	ECL ³⁴
Estandares de eLearning soportados	OKI JSR 168 (Java Portlet Specification)
Idiomas	Inglés
LCMS	No
Extras	<ul style="list-style-type: none"> - Herramienta de notificación (anuncios) - Chat - Email - Mensajería - Espacio de trabajo - Agenda/Calendario - Tests - WebDAV - Wiki
Notas	

³³ <http://www.sakaiproject.org/>

³⁴ Educational Community License Version 1.0. <http://sakaiproject.org/license.html>

Cuadro comparativo												
	LMS >	1	2	3	4	5	6	7	8	9	10	Total
L E N G U A J E	J2SE						√			√	√	3
	J2EE				√							1
	PHP		√	√		√		√	√			5
	TCL	√										1
B A S E S	ORACLE	√			√					√	√	4
	MySQL		√	√	√	√	√	√	√	√	√	9
	PostgreSQL	√			√					√		3
	HSQLDB				√					√		2
	Otras				√							1
I	Idiomas	1	>15	>30	1	>30	2	>15	>40	8	1	≈14*
C	LCMS	√	√			√		√		√		5
E S T A N D A R E S	SCORM 1.2	√	√	√ ²		√		√	√	√		6-7
	IMS QTI				√ ⁴			√	√	√ ⁵		4
	IMS CP	√	√	√ ³	√	√		√	√ ³	√		8
	IMS LD	√ ¹			√							2
	IMS ePortfolio				√							1
	JSR 168										√	1
	W3C WCAG 1.0		√									1
	W3C XHTML 1.0		√		√							2
F U N C I O N A L I D A D	Foro	√		√	√	√		√	√			6
	Chat	√	√	√	√	√		√	√	√	√	9
	Agenda/Cal.	√	√		√		√	√		√	√	7
	Encuestas	√				√	√			√		4
	Galería Fotos	√	√					√				3
	Tablón Noticias	√							√		√	3
	Repositorios	√	√						√			3
	Búsquedas	√				√						2
	Tests	√	√	√		√		√			√	6
	Blogs	√	√									2
	VideoConf		√			√			√			3
	Personalizable		√		√				√	√	√	5
	Mensajería				√				√	√	√	4
	Almacenamiento	√			√		√			√	√	5
	Wiki		√						√	√		3
	CMAP		√									1
Text to Speech		√									1	

Leyenda

Primea Fila: 1: .LRN; 2: Atutor; 3: Claroline; 4: CooperCore; 5: Dokeos;
6: Eledge; 7: Ilias; 8: Moodle; 9: OLAT; 10: Sakai

¹ Sólo Niveles A y B

² No conforme con SCORM RTE

³ Sólo permite importar cursos, pero no exportar.

⁴ Implementa la versión 2 de IMS QTI

⁵ Implementa la versión 1.2.1 de IMS QTI

* Media aritmética aproximada

Como se puede observar en el cuadro comparativo anterior, si nos atenemos a los lenguajes de programación empleados para el desarrollo de estas herramientas, PHP (5) es el más usado seguido de Java (4).

En cuanto a la persistencia de los datos, todos usan al menos una base de datos. El 40% de las herramientas analizadas admite configuración de la base de datos, aceptando dos o más bases de datos diferentes. Destacan en este aspecto CopperCore y OLAT.

Algo que es deseable en todas las herramientas destinadas a la educación es que independientemente de los contenidos, los datos que estas herramientas presenten al usuario (ayuda, controles, interfaz gráfica, etc) pueda ofrecerse en varios idiomas de entre los que el usuario puede elegir su preferencia. Esto es así en 7 de los 10 casos analizados, siendo 3 ellos monoidioma.

El 50% de los LMS estudiados ofrece la funcionalidad necesaria para crear cursos, contando como en el caso de OLAT con un almacén de cursos dónde los diseñadores pueden subir y registrar dichos haciéndolos accesibles a los usuarios según criterios de nivel de dificultad, por grupos, etc.

SCORM se perfila como la especificación de eLearning más ampliamente aceptada entre las herramientas mostradas, siendo por tanto IMS CP la especificación preferida a la hora de empaquetar los contenidos. IMS Learning Design apenas cuenta con LMS aún excepto la iniciativa de la UONL, CooperCore. Cabe destacar también que el 40% de las herramientas de gestión del aprendizaje, soportan la especificación IMS QTI para la evaluación a través de tests.

Entre la funcionalidad extra ofrecida por los LMSs descritos, cabe destacar que casi todos (9 de los 10 LMS descritos) cuentan con un servidor de chat o mensajería instantánea que ofrece la posibilidad de intercambiar opiniones mediante clientes propios o de terceros compatibles con su servidor (Jabber por ejemplo). Otras funcionalidades ampliamente aceptada y ofrecida por este tipo de herramienta son las agendas y calendarios, que permiten a los usuarios anotar eventos de interés e incluso recibir notificaciones de alarma para los eventos elgidos; foros, y la posibilidad de realizar tests.

De entre todos los LMSs mostrados en el punto anterior, .LRN y ATutor son los que más funcionalidades extras ofrecen. Atutor destaca además por ofrecer extras no muy comunes como son: comunicación mediante videoconferencia, blogs de usuarios, cliente de creación y edición de mapas conceptuales (CMAP), Wiki o la posibilidad de audio-descripción del texto para alumnos con deficiencias visuales.

Otras herramientas que han tenido una gran repercusión han sido los editores de meta-datos y reproductores de cursos que permiten la creación y edición de cursos siguiendo diferentes especificaciones de empaquetado (incluidas las diferentes versiones de una misma especificación) en el caso de los primeros, y las pruebas y validación de estos en el caso de los segundos.

En este tipo de herramientas han tenido gran repercusión las desarrolladas por Reload³⁵ que incluyen un editor de metadatos que permite diferentes perfiles de aplicación, un creador de cursos empaquetados (con IMS o con SCORM), un editor de IMS Learning Design y visualizadores (players) para que se pueda ver el resultado obtenido con las herramientas. Hay herramientas que soportan IMS QTI, fundamentalmente comerciales como QuestionMark³⁶ o CanvasLearning³⁷.

Otras herramientas más generalistas, aunque igualmente utilizadas para la formación, como BSCW³⁸ y GForge³⁹, permiten la colaboración en tiempo real y de forma colaborativa, incluyendo entre su funcionalidad la catalogación de proyectos (o cursos), subir trabajos, crear foros para determinadas áreas, etc.

Otros sitios web para encontrar información sobre las últimas herramientas compatibles con los estándares son Academia ADL Co-lab⁴⁰ y los sitios web de ADL e IMS. Además otros sitios de referencia para mantenerse al día de las continuas evoluciones de los estándares son el Centre for Educational Technology Interoperability Standards⁴¹, el Learning Technology Standards Observatory⁴² del Centro Europeo para la Normalización, así como los diferentes observatorios tecnológicos creados en España para tal fin:

³⁵ <http://www.reload.co.uk>

³⁶ <http://www.questionmark.com>

³⁷ <http://www.canvaslearning.com>

³⁸ <http://bscw.fit.fraunhofer.de/>

³⁹ <http://gforge.org/>

⁴⁰ <http://www.academiccolab.org>

⁴¹ <http://www.cetis.ac.uk>

⁴² <http://www.cen-ltso.net>

Nombre	Propietarios	URL
Observatorio Tecnológico	Ministerio de Educación y Ciencia	http://observatorio.cnice.mec.es/
Pulsar - Observatorio e-learning	Universidad del País Vasco	http://pulsar.ehu.es/
Observatorio Galego de e-learning	Xunta de Galicia, Univ. De Santiago de Compostela, Univ. de Vigo, Univ. Da Coruña, CESGA, EOSA	http://observatorioel.cesga.es/
Observatorio de e-learning	Universidad de Oviedo	http://www.elearning.uniovi.es/
Observatorio de e-learning	Universidad Politécnica de Madrid	http://madeira.ls.fi.upm.es/o-e-learning/index.jsp
Portal Andaluz de e-learning	Junta de Andalucía	http://prometeo3.us.es
Gabinete de Tele-Educación	Universidad Politécnica de Madrid	http://www.gate.upm.es/plataformas/plataformas_vistas/index.htm

Tabla 5. Lista de webs *observatorio* del estado del eLearning en España

2.7.La plataforma tecnológica de eLearning

En español se ha denominado Plataforma Tecnológica (en inglés LMS: “Learning Management System”) a un sistema informático que ofrece los servicios de gestión del proceso de aprendizaje en cursos a distancia: acceso a los cursos y sus materiales didácticos, herramientas de comunicación (correo electrónico, foros, chats), seguimiento de la evolución de los alumnos, almacén de recursos didácticos (biblioteca digital, videoteca, enlaces externos...), etc. Como acabamos de ver, algunos sistemas cuentan también con la funcionalidad asociada a la creación, edición y publicación de cursos (en inglés LCMS: “Learning Content Management System”).

Sin embargo uno de los principales problemas es la escasa interoperabilidad entre los distintos LMS, incluso en los de código abierto. Esto es debido principalmente a que a pesar de que la mayoría de ellos implementan una o más especificaciones de las ampliamente aceptadas, no todos implementan las mismas para la misma funcionalidad e incluso algunos optan por formatos propietarios rompiendo intencionadamente la compatibilidad. En cuanto a la duda entre optar por aplicaciones que implementan estándares abiertos pero de código propietario o aplicaciones de código libre, J. Dalziel apunta en [22] que son las aplicaciones comerciales las que se ven más obligadas a cumplir estándares abiertos para asegurar el éxito de éstas, puesto que en las aplicaciones de código abierto, siempre cabe la posibilidad de a partir del código, realizar las modificaciones oportunas para cumplir los estándares necesarios (a menudo dando lugar a nuevos productos).

Para ilustrar esta falta de entendimiento a pesar las especificaciones existentes para garantizar la interoperatividad podemos citar el caso de la especificación IMS DRI (Digital Repositories Interoperability) creada precisamente para garantizar la interoperatividad entre almacenes de recursos digitales. El proyecto OKI está basado en APIs de código abierto e incluye un servicio de almacén de recursos digitales. Este proyecto estuvo en estrecha colaboración con el consorcio IMS durante el mismo periodo en que la especificación DRI fue desarrollada. Sin embargo OKI, a pesar de contar con una financiación importante, alta publicidad y una estrecha colaboración con IMS, eligió su propio formato de almacenes de recursos digitales. Curiosamente el Centro de Investigación de e-learning de la Universidad de Cambridge (CARET) intentó implementar las dos aproximaciones, IMS DRI y el formato de OKI, y la conclusión final resulto ser: “era más difícil de lo que esperábamos” [22].

2.7.1. Arquitecturas de los sistemas de enseñanza a distancia

Casi todas las propuestas optan por arquitecturas cliente-servidor, algunas de ellas cuyos modelos funcionales concreticen el estándar LTSA con las especificaciones existentes (empaquetamiento de contenidos, información sobre el perfil del usuario, etc) hasta el momento. Un ejemplo de ello se proporciona en [23], dónde se muestra una arquitectura *implementable* de un sistema e-learning en el que, como se verá a continuación, se pueden agrupar ciertos módulos funcionales de forma similar a las entidades de LTSA.

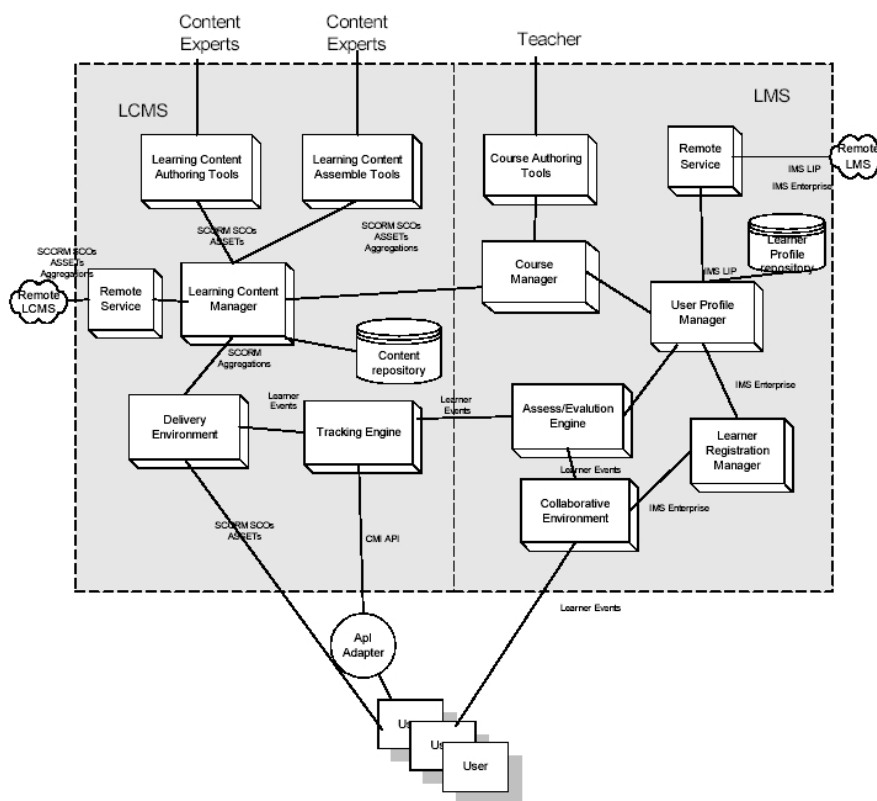


Figura 14. Modelo funcional del Sistema eLearning de Liu, El Saddik y Georganas

Si eliminamos la entidades correspondientes con las herramientas de autoría de la figura anterior, podemos identificar algo muy parecido a la arquitectura LTSA en el modelo funcional anterior:

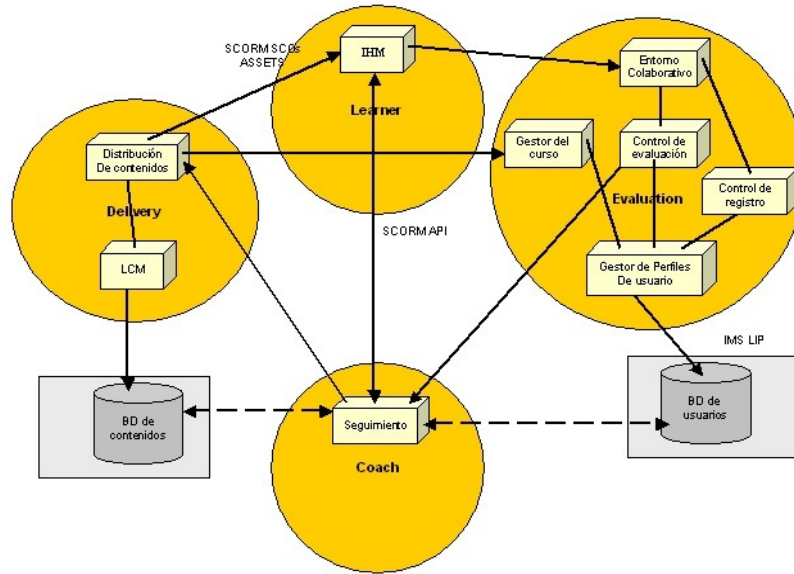


Figura 15. LTSA presente en el modelo funcional de Liu, El Saddik y Georganas.

En este modelo funcional, según los diagramas de las dos figuras anteriores, a diferencia del diagrama de LTSA, el módulo de Seguimiento (Tracking Engine en la primera de las dos figuras) no accede directamente al almacén de datos que contiene la información del usuario (perfil, expediente académico, etc) sino que delega esta funcionalidad en el módulo funcional "Control de Evaluación" (Assess/Evaluation Engine en la primera figura). Esta es una de las mínimas diferencias la propuesta de Liu y el estándar. De igual forma, este módulo de Seguimiento del proceso de aprendizaje tampoco obtiene flujo de datos de forma directa del almacén de contenidos en el modelo de la figura anterior, sino que las particularidades de los lenguajes de acceso y consulta a este almacén de datos se abstraen de dicho módulo funcional y se delegan en el de distribución de contenidos.

Aunque no todas las implementaciones de sistemas eLearning toman como referencia el estándar de IEEE, sí que coinciden la mayoría en la necesidad de garantizar la interoperatividad con otros sistemas independientemente del lenguaje de programación utilizado o de la arquitectura interna de cada uno. Por ello muchos de estos sistemas de eLearning optan por arquitecturas orientadas a servicios, puesto que estos permiten exponer la funcionalidad de una aplicación a través de interfaces que pueden ser accedidas a través de la red. Además los servicios pueden ser orquestados con el fin de construir y ofrecer nuevas funcionalidades más complejas, abriendo así una nueva vía para la colaboración entre diferentes plataformas de eLearning. Así tanto en la arquitectura del modelo funcional del ejemplo anterior como en otras propuestas [24], [25] se publica como servicio la funcionalidad que potencialmente puede ser utilizada por otras plataformas.

Las arquitecturas orientadas a servicios promueven el diseño de aplicaciones como un conjunto de elementos poco acoplados que interactúan entre sí, llamados servicios. Como definición de servicio podríamos decir que "es una entidad que puede ser usada por una persona, programa u otro servicio, que realiza funciones de cómputo, de almacenamiento o de comunicación con otros usuarios, dispositivos físicos o incluso con otros servicios" [31].

Al margen del soporte que dan las diferentes tecnologías al concepto de servicio, una arquitectura orientada a servicios se organiza de acuerdo a ciertas reglas bien definidas: los servicios deben tener interfaces descritas mediante un lenguaje formal que defina las operaciones que puedan solicitar otros servicios. Para completar el dinamismo del modelo es además necesario que

los servicios puedan ser publicados en registros accesibles mediante mecanismos estandarizados, que permitan su descubrimiento y su posterior utilización a otros servicios o usuarios.



Figura 16. Arquitecturas Orientadas a Servicios.

Adquieren relevancia en este tipo de arquitecturas los proveedores de servicios y los almacenes (tanto de servicios como de contenidos) con los que estos proveedores trabajan. Para ello en la propuesta de arquitectura de Liu, El Sadik y Georganas [23] se definen mecanismos de acceso a servicios remotos, si bien no aparece explícitamente la figura del proveedor de servicios. Estos servicios remotos serían los encargados de asegurar la interoperatividad para compartir contenidos gestionados por LCMS remotos. También justifica el uso de una arquitectura orientada a servicios para asegurar la interoperatividad entre LMS independientes física y tecnológicamente a la hora de compartir perfiles de usuarios, de cara a facilitar la federación de identidades y favorecer el acceso y uso de recursos de “LMS amigos”.

Existen distintas tecnologías que pueden ser catalogadas dentro del paradigma de las Arquitecturas Orientadas a Servicios, entre las que cabe mencionar: Servicios Web⁴³, OSGi⁴⁴ o Jini⁴⁵. Cada una de ellas tiene un ámbito de aplicación y objetivos diferentes; por ejemplo, Jini se orienta a la publicación de servicios en redes locales y de corto alcance, los Servicios Web tratan de definir las interacciones entre servicios disponibles en Internet, y OSGi busca la integración de los servicios domésticos, de área local y personal con Internet. El principal objetivo de la alianza OSGi, es definir un conjunto de especificaciones software abiertas que permitan diseñar plataformas compatibles que puedan proporcionar múltiples servicios, de ahí que esta alianza esté formada por multinacionales principalmente relacionadas con la electrónica y las telecomunicaciones como: IBM, Intel, Mitsubishi Electric, NEC, NTT, Siemens, Motorola, Nokia, Samsung, Telefonica I+D, BEA, Hitachi, Ericsson o Robert Bosch. Inicialmente OSGi fue pensado principalmente para su aplicación en redes del hogar y por ende en la llamada domótica o informatización del hogar, pero su uso se ha ido extendiendo a gran variedad de sistemas empotrados, desde aplicaciones en automoción a telefonía móvil, lo que ha repercutido en el registro de OSGi como marca registrada (que servirá para certificar los productos conformes a sus especificaciones) perdiendo estas siglas el significado primitivo relacionado con las pasarelas residenciales de servicios abiertos (Open Services Gateway)

Aunque OSGi define su propia arquitectura, ha sido pensada para su compatibilidad con Jini o UpnP.

En la arquitectura de OSGi adquiere especial relevancia el concepto de Plataforma de Servicios, situada en la red local y conectada al proveedor de servicios a través de una pasarela en la red del operador. Este elemento será el responsable de permitir la interacción entre dispositivos o redes de dispositivos que podrían utilizar distintas tecnologías para comunicarse.

⁴³ Servicios Web, <http://www.w3.org/2002/ws/>

⁴⁴ Open Services Gateway Initiative. OSGi Alliance. <http://www.osgi.org/>

⁴⁵ Jini Network Technology. <http://www.sun.com/software/jini/>

Cabe destacar en este estudio sobre el estado del arte relacionado con las tecnologías para eLearning, dos enfoques llamativos como son: el uso de redes punto a punto (P2P⁴⁶) y los portales web del estilo de Wikipedia⁴⁷.

En las redes P2P, los clientes usados permiten la búsqueda de contenidos en el resto de nodos, con lo que en esta arquitectura todos los nodos actúan simultáneamente como clientes y como servidores. Estos clientes catalogan los contenidos almacenados en su propio nodo, generalmente en base al tipo de archivo (audio, video, documentos PDF o Word, páginas HTML, imágenes en JPG, etc) pero también es posible usar los metadatos para catalogar y clasificar los contenidos según el nivel del alumnado al que va dirigido, el idioma, la materia o asignatura relacionada, etc. Un ejemplo de este enfoque es el proyecto Edutella, un sub-proyecto incluido dentro del proyecto PADLR⁴⁸ (Personalized Access to Distributed Learning Repositories) centrado en el desarrollo de una infraestructura básica, herramientas e interfaces para el uso de recursos distribuidos, haciendo uso de una red P2P para compartir recursos educativos. Edutella usa RDF para como lenguaje de descripción de los recursos disponibles añadiendo de esta forma información semántica a los metadatos.

Wikipedia representa un enfoque tecnológico diametralmente opuesto a Edutella pero pedagógicamente afín, pues son los propios usuarios quienes construyen, corrigen y validan los contenidos. Wikipedia ha tenido un gran impacto en la comunidad educativa y el porcentaje de alumnos que la usan tanto para introducir y editar contenidos como para obtener información no ha parado de crecer desde su creación. Ha sido muy criticada desde sus inicios al no contar con un grupo de revisores especializado, posibilitando la introducción de información errónea (ya sea intencionadamente o no) que los alumnos no se preocupan de contrastar, dada la credibilidad que otorgan a las TIC, llegando a ser vetada por algunos centros educativos como fuente de información para los trabajos realizados por sus alumnos.

⁴⁶ *Peer To Peer*

⁴⁷ Wikipedia. <http://www.wikipedia.org>

⁴⁸ Personalized Access to Distributed Learning Repositories - Final Proposal.
<http://www.learninglab.de/workspace/padlr/index.html>

3. Planteamiento del problema

3.1. Introducción

Hemos repasado el estado del arte del eLearning en cuanto a especificaciones, estándares, y herramientas se refiere, sin profundizar en las particularidades asociadas al medio de distribución o la forma de presentar los contenidos e interactuar con el usuario, lo que nos llevaría a diferenciar diferentes tipos de eLearning como t-learning o m-learning.

Esta diversidad de medios de distribución y los dispositivos usados como receptores (televisores digitales, teléfonos móviles, PDAs, etc.) hace necesaria la conversión de formato de un mismo contenido para que pueda ser visualizado de forma eficiente y adecuada a las características de cada dispositivo. En estos tiempos en los que la globalización y la dependencia de la Red se acentúan cada vez más, un acceso flexible a la información, independientemente del lugar, del tiempo y de los dispositivos de comunicación usados, marcará la diferencia a la hora de aumentar la competitividad y productividad así como reducir los costes dedicados a la formación. [22]

La implantación obligatoria de la TV digital y consecuente “apagón analógico” (en 2010 en España y en Europa a partir de 2012), la telefonía móvil, los ordenadores de bolsillo, la automoción, etc., introducen un nuevo grado de complejidad al problema de interoperatividad visto para diferentes sistemas de eLearning. No sólo han de buscarse soluciones para asegurar la interoperatividad entre dispositivos, sino también soluciones que permitan el despliegue y descubrimiento de aplicaciones en diferentes dispositivos de forma transparente para el usuario, posibilitando la instalación y configuración de dichas aplicaciones en función de las necesidades del usuario y del contexto en que se encuentra. Nos encontramos frente a problemas de adaptación de contenidos y conciencia del contexto (context awareness).

Se consideran “adaptaciones del contenido” aquellas, que modifican parcial o totalmente el contenido mostrado al usuario según su perfil, sus preferencias, etc., pero no supone un cambio en los mecanismos ni en el formato en que dicho contenido es proporcionado al usuario. Por ejemplo: adaptaciones del contenido de una página HTML para mostrarlo en el idioma preferido del usuario.

Por su parte la definición de contexto es algo más amplia e imprecisa, y debe responder a tres preguntas principalmente: “¿dónde estás?”, “¿con quién estás?” y “¿qué recursos tienes alrededor?” [28]. La expresión “context awareness” o conciencia del contexto fue introducida por Schilit [29] en 1994 para referirse, expresamente en el campo informático de los sistemas distribuidos, al hecho de que los dispositivos puedan simular que perciben lo que ocurre a su alrededor, sienten y son capaces reaccionar en base al entorno que les rodea, simulando así esa conciencia acerca del contexto en el que se encuentran. Este contexto engloba algo más que simplemente la ubicación del usuario, pues puede haber factores variables como luces (encendidas o apagadas), el coste de las comunicaciones en una determinada franja horaria, (para facturar con la empresa más barata en cada momento), el nivel de ruido, las conexiones de red inalámbrica existentes y sus características, etc. Chen y Kotz [30] definen el contexto como “el conjunto de estados del entorno que determinan el comportamiento de una aplicación o en qué eventos se producen cambios interesantes para el usuario”. Ejemplos de “context awareness” relacionados con el eLearning, pueden ser:

–Un teléfono móvil puede detectar cuando está en el aula y cuando el usuario está sentado, concluyendo que el usuario está actualmente en clase y rechazando automáticamente cualquier llamada o silenciando la señal sonora de llamada.

–Tenorio S.A., es una empresa de ingeniería que tiene la capacidad de incorporar innovaciones y mejoras muy rápidamente en las máquinas que fabrica. Don Juan es un experto comercial en conquistar a los clientes y en vender las nuevas máquinas, y esto hace que ocupe más de tres cuartas partes de su calendario laboral viajando para visitar a los clientes, con la intención de hacerles renovar la maquinaria antigua o conquistar nuevos clientes. Para ello Don Juan necesita conocer perfectamente las especificaciones técnicas tanto de las nuevas máquinas para tratar de convencer al cliente las ventajas sobre la anterior. Pero debido al gran número de viajes que Don Juan debe hacer, se ve obligado a estudiar las características durante los propios viajes. Pero Tenorio S.A. ofrece a Don Juan una herramienta muy potente que le permite procesar los datos de las especificaciones, obtenidos desde el Servidor Central de Especificaciones Técnicas, y en función del medio de transporte adaptarlo a sus necesidades. Así cuando viaja en tren o en avión, puede usar su ordenador portátil para leer las especificaciones, cuando lo hace en autobús o en taxi puede exportar la especificación a un formato especial para visualizarlo en su NOKIA 770 y cuando conduce él es informado de un resumen de las características principales a través de una aplicación TTS⁴⁹ que lee dicho resumen por él.

–Aplicación diccionario: Además Don Juan realiza mucho de sus viajes al extranjero, y aunque tiene un muy buen nivel de inglés, en muchos viajes se ha visto forzado a elegir el menú al azar por no contar el restaurante con personal que hable español o inglés, y no contar con una traducción de la carta a alguno de los idiomas que Don Juan domina. Pero no hay problema, porque la aplicación diccionario que Don Juan tiene instalada en su móvil, se adapta al país en el que se encuentra y Don Juan sólo necesita escribir la palabra solicitada y el diccionario le devuelve su traducción a español, el idioma nativo de Don Juan.

La conjunción de adaptabilidad y conciencia del contexto, le confiere al entorno una especie de “inteligencia”, que desde el ISTAG⁵⁰ de la Comisión Europea se comenzó a denominar “Inteligencia Ambiental”.

Esta propuesta de investigación propone aplicar técnicas de adaptación de contenidos y conciencia del contexto en especificaciones eLearning en pro de la consecución de un middleware para eLearning práctico, fácil de usar, que permita acceder en cualquier momento, desde cualquier lugar y de cualquier forma posible a un aprendizaje personalizado de alta calidad. Partimos para ello de los avances obtenidos a partir de los proyectos OSMOSE y Passepartout, que se resumen en el siguiente apartado de esta memoria.

El trabajo de investigación persigue combinar OSGi y las especificaciones de eLearning descritas en el estado del arte de esta memoria, definiendo una arquitectura orientada a servicios, abierta, que posibilite diferentes modos de uso para favorecer tanto el aprendizaje on-line como el aprendizaje off-line (usando dispositivos móviles como PDAs o teléfonos móviles) y facilitando la incorporación de plug-ins que permitan añadir la funcionalidad necesaria para lograr una plataforma capaz de adaptarse a las necesidades del usuario en pro de un aprendizaje personalizado.

3.2.Relevancia

Presentamos a continuación algunos puntos que nos ayudarán a tener una mejor idea de la importancia de este tipo de investigaciones.

⁴⁹ Text To Speech

⁵⁰ Information Society Technologies Advisory Group. <http://cordis.europa.eu/ist/istag.htm>

3.2.1.Desde el punto de vista económico

- Como ya hemos visto, reduce los costes económicos en el desarrollo de aplicaciones de eLearning.
- El impacto de Internet está siendo tan importante que incluso está quitando audiencia a los canales de TV tradicionales. Cada vez hay más personas (sobre todo las más jóvenes) que prefieren el ocio de Internet (juegos en red, navegar, chatear...) y se informan de la actualidad a través de los medios de información de Internet antes que por televisión. Aprovechando el tirón de la Red, algunos programas de televisión se limitan a mostrar vídeos obtenidos de páginas web o que los mismos telespectadores envían por correo electrónico, muchos de ellos con una calidad lamentable, para entretener a la audiencia. La red engancha y tiene una gran repercusión en la sociedad actual. Todo esto influye en la publicidad y dispara el interés de los anunciantes por estar presentes en la red de redes, un medio que les ofrece una audiencia potencial superior a la de la televisión y a un coste mínimo. Esto ha potenciado nuevas prácticas de publicidad (como la publicidad viral) como también está potenciando nuevas prácticas de formación y aprendizaje.
- Con la aparición de la TV digital, ha aumentado espectacularmente la venta de pasarelas residenciales en detrimento de los set-top box convencionales (sólo descodificadores de TV digital) y algunos estudios se aventuran a pronosticar un descenso proporcional en la venta de ordenadores y videoconsolas, si las pasarelas residenciales integran ambas funciones. Muchas de estas pasarelas no sólo se limitan a decodificar la señal digital de televisión, sino que ofrece otros servicios que aumentan su valor añadido: reproductor de DVD, reproductor de MP3, DIVX, lector de tarjetas SIM, SD, bluetooth, etc.
- No es descabellado pensar que esta irrupción en el mercado de nuevas pasarelas residenciales de lugar a nuevos modelos de negocio, más orientados a los servicios: desarrollador de servicios, agregadores de servicios, gestores de pasarelas, gestores de la red de acceso, del centro de control, etc.
- Bajo coste a largo plazo. Una vez que las pasarelas residenciales vayan ganando cuota de mercado, su precio será muy bajo (muy inferior al de un PC convencional).

3.2.2.Desde el punto de vista legal

- En cuanto al uso de una pasarela residencial como plataforma tanto para el procesado de la señal de TV digital como para el despliegue de aplicaciones educativas, es necesario constatar que la televisión digital está regulada por leyes españolas y directivas europeas, siendo obligatoria la emisión en formato digital en Europa a partir del 1 de Enero de 2012. En España, se adelantó el llamado apagón analógico al 1 enero de 2010, mediante el Real Decreto del 29 de Julio de 2005 al (excepto en Soria, que sucederá en 2008 y que servirá de prueba piloto).
- Tanto la Legislación europea como la española animan al desarrollo de servicios digitales avanzados para TV digital.
- La Comisión Europea anima al desarrollo de proyectos educativos basados en eLearning y mantiene distintos foros especializados en Internet así como distintos programas de financiación de estos proyectos (programa marco, programa IST...)
- Los proyectos de domótica cada vez están teniendo más aceptación entre los usuarios, haciéndose más accesible para todos. Integrar un sistema educativo en una pasarela

residencial totalmente compatible con cualquier aplicación domótica proporciona innumerables ventajas y una mayor aceptación social.

- Al utilizar sólo software libre, no sujeto a patentes, se facilitará su uso en centros de enseñanza.

3.2.3.Desde el punto de vista social

- Si tenemos en cuenta la influencia que ha tenido la telefonía móvil en la sociedad en los últimos años, la integración de ésta en un proyecto destinado a la educación no parece muy lejos.
- Podrían crearse canales temáticos de TV digital interactiva, centrados en la educación. Por ejemplo el canal temático US (Universidad de Sevilla) en el que se proporcione acceso bajo demanda a archivos audiovisuales de alta calidad, retransmisiones de eventos y acontecimientos de la US (deportes, congresos, seminarios,...) y establecer tarifas para cada uno de ellos, videoconferencias, servicios de información (mensajería SMS, MMS...), seminarios semipresenciales vía satélite, etc.

3.2.4.Desde el punto de vista tecnológico

- Desde el tercer trimestre de 2002 las cadenas españolas Telecinco, Antena 3 y Canal+, emiten simultáneamente en formato digital, y desde 2006 todas las cadenas nacionales y regionales de televisión.
- El poder de su arquitectura radica en los servicios. Es posible desarrollar cualquier tipo de servicio (con cualquier funcionalidad) que pueda instalarse en la pasarela.
- OSGI incorpora un servicio de publicación WEB, haciendo posible el acceso a servicios que tengamos instalados en la pasarela, de forma que el alumno puede acceder cuando quiera y desde cualquier punto a los documentos que publique.
- OSGI está pensado para soportar la interconexión con otros dispositivos, de forma que la pasarela se convierte en el nodo centralizador de las comunicaciones entre los dispositivos del hogar entre sí, así como entre éstos y el exterior.
- Se trabajarán con un gran número de tecnologías atacando problemas que aún no han sido resueltos, resultando en un proyecto de alto valor tecnológico e innovador.

3.2.5.Desde el punto de vista de la enseñanza

- Las aplicaciones educativas serán de mayor calidad. Podrán ofertarse la retransmisión de seminarios, conferencias, clases magistrales, etc.
- Supone una ventaja importantísima para los estudios relacionados con profesiones artísticas y técnicas: Bellas Artes, Arte Dramático, Arquitectura... al poder ofrecerse con gran calidad contenidos multimedia interactivos relacionados directamente con las obras.
- Acorta las distancias alumno/centro, alumno/profesor.

- Supone un nuevo campo de experimentación en cuanto a metodologías de aprendizaje, pues se pretende una participación interactiva del alumno ante un medio en el que tradicionalmente el receptor de la información ha sido un actor pasivo.
- La creación de canales educativos personalizables constituye un objetivo interesante.
- Educación a la carta, mediante la combinación de la programación de diversos canales educativos.

3.2.6.Desde el punto de vista de la investigación

Desde el punto de vista de la investigación este trabajo ha contribuido a la publicación de artículos enfocados en temas relacionados más concretos, como son la propuesta de una plataforma de servicios abiertos en un entorno domótico [31], la definición de un mecanismo de despliegue remoto de componentes en plataformas de servicios OSGi [35] así como herramientas para la resolución de dependencias entre unidades de despliegue [40], la definición e implementación de un catálogo de servicios E2E para plataformas de servicios OSGi [36] [41], la abstracción entre componentes de navegación y contenidos [42] y por supuesto en el ámbito de la docencia [43] [44] [45].

El autor ha colaborado con otros investigadores en algunas áreas más concretas, relacionadas con posibles escenarios de uso de la plataforma descrita en este trabajo, como la citada anteriormente relativa al despliegue de componentes, de la que además de los artículos citados, ha sido escrita una Tesis [46] de la cual el proyecto de Tesis a partir de esta memoria de investigación puede beneficiarse.

Por otra parte, en cuanto a la la integración de tecnologías para conseguir escenarios multimedia en los que se lleven a cabo actividades educativas, la participación del autor y gran parte del contenido de este trabajo en el proyecto Paseepartout, ha contribuido a asociar el concepto de SCO y bundle como algo similar, ha servido para definir un modelo de despliegue de componentes así como herramientas para la gestión de dicho despliegue [40] y ha contribuido a crear un grupo de usuarios de OSGi en España⁵¹. Es además un campo de investigación que está despertando gran interés en las compañías del sector audiovisual, como muestran los avances del proyecto Passepartout (en el que el autor de este trabajo participó) dónde este trabajo complementó una muy interesante propuesta de Philips Applied Technologies: AMP⁵² (Ambient Multimedia Platform) que integra diferentes tecnologías como TVAnytime, BlueRay, Java 3D y detección de movimiento usando cámaras web en una mesa, para construir un escenario inspirado en la película Minority Report, y que puede crear ambientes para favorecer la comprensión de conceptos (favoreciendo en definitiva el aprendizaje) a través tanto de los contenidos multimedia como de las sensaciones y emociones creadas por estos ambientes multimedia a través de sus actuadores X10 para controlar la intensidad de la luz, subir o bajar persianas, controlar la temperatura de la sala...

En este trabajo se aportan ideas para la aplicación de las arquitecturas orientadas a servicios y el despliegue de componentes a la distribución de cursos de eLearning, lo que supone un enfoque novedoso puesto que el enfoque más extendido actualmente siguen siendo las tradicionales aplicaciones web, con arquitectura cliente-servidor, de cliente ligero (página web, applet o flash) que debe ser descargado bajo demanda y ejecutado desde el navegador del usuario. Con este enfoque se posibilita, además de lo anterior, la construcción de clientes pesados que puedan ser idóneos para otro tipo de aprendizajes (simuladores, entrenadores, etc.) siguiendo una arquitectura orientada a servicios, escalable y abierta.

⁵¹ OSGI Users' Forum. Spain. <http://spain.osgiusers.org>

⁵² Puede verse un vídeo demostración en: http://www.youtube.com/watch?v=tsLjGyZ_RCs

3.3.Problemas existentes

Basándonos en las características de los LMS comentados en el apartado anterior podemos deducir algunos inconvenientes de estos que pretendemos solucionar con la investigación posterior:

- La mayoría de los LMS actuales están orientados principalmente a soportar cursos SCORM 1.2 (pocos soportan actualmente SCORM 2004), añadiendo funcionalidad adicional como chats (90% de los mostrados en el estado del arte), agendas y calendarios (el 70%), foros (60%) y ya en menor medida algunos permiten la personalización de la interfaz gráfica, el uso de software de mensajería instantánea, etc.
- En muchos de los LMS basados en la Web, el cliente se compone además del navegador web, de una aplicación cliente que es desplegada en la máquina del usuario en el momento de cargar la página (en algunos casos, si no ha habido modificaciones, sólo la primera vez que se carga). Esto introduce además de la dependencia del navegador (que realmente no supone ningún inconveniente), la dependencia de la tecnología concreta que dicha aplicación necesita: JRE si se trata de applets, el plug-in apropiado para el navegador si se trata de Flash o Shockwave, y esto si que puede suponer un problema de no existir la tecnología necesaria para el hardware, el Sistema Operativo o el navegador usado por el dispositivo cliente.
- Además de la dependencia de un navegador y en algunos casos de una o más tecnologías concretas (Java, Flash, Showckwave...) en la mayoría de los casos sólo es posible el avance en el curso en modo on-line. No se permite el avance off-line que sería factible mediante el almacenamiento temporal de los items superados y su posterior sincronización con el LMS remoto del servidor una vez se recupere la conexión (sincronización automática) o bien a petición del usuario (sincronización bajo demanda). Esto sería muy útil a determinados profesionales como nuestro Juan de Tenorio S.A. que necesitan una formación rápida *in mision*.
- Sólo la mitad de los LMS permiten la personalización del entorno por parte del usuario, independientemente del rol de éste. La mayoría entiende por personalización sólo la modificación del aspecto gráfico mediante el uso de diferentes estilos, usando hojas de estilo CSS u otros mecanismos similares para crear temas que el usuario incluso puede editar y personalizar a su gusto, pero sólo un pequeño porcentaje de estos permite una personalización total, orientada a plug-ins que permita instalar nuevas aplicaciones (organizadores, calculadoras, reproductores, aplicaciones de videoconferencia...) y módulos que aumenten la funcionalidad extra aportada por el LMS.
- SCORM establece como mecanismo de comunicación entre el entorno del cliente y el LMS un API implementado en el lenguaje estándar ECMAScript (más conocido por javascript). Creemos que esta concreción es en sí un inconveniente y que debería ser independiente del lenguaje en el que se implemente. Por ejemplo, esta restricción referente al uso de javascript dificulta la implementación de aplicaciones conformes con la especificación de SCORM para dispositivos que no cuentan con navegadores son soporte para javascript. Igualmente dificulta la implementación en Java de clientes del LMS con interfaz gráfica (ya sea Swing, SWT, etc.) y sin usar un navegador web, dada la fuerte dependencia de SCORM con respecto a javascript (especialmente negativa es esta dependencia para la implementación de clientes del LMS en pequeños dispositivos móviles).
- Hay una ruptura metodológica entre las educación mediante clases presenciales (especialmente en la formación reglada) y la tele-formación, debido no sólo a las diferencias entre los mecanismos de interacción social de una y otra, sino al enfoque pedagógico: fuertemente conductista en SCORM y mayoritariamente constructivista en la docencia presencial. Además SCORM no favorece otras metodologías, de ahí el auge que está experimentando en los últimos años IMS Learning Design.

- La especificación de SCORM obliga a incluir la API (implementada en javascript) de comunicación con el LMS en cada paquete SCORM, con lo que en caso de implementar una plataforma para reproducción de cursos off-line, capaz de almacenar localmente los cursos, este API estaría repetida en todos los paquetes. Sería más eficiente publicar esta API de comunicación en alguna ubicación externa al paquete SCORM, eliminando esta multiplicidad del código. Esto implica que los contenidos del paquete SCORM han de saber su localización o disponer de un mecanismo para localizarla. En definitiva, otra API. El mismo problema. Sin embargo, las arquitecturas orientadas a servicios pueden ser de ayuda para este tipo de problemas. La solución pasa por especificar abiertamente un servicio encargado de comunicar y usar el API, de hacer de proxy del SCO. Es aquí dónde cobra especial relevancia el uso de frameworks de servicios como OSGi o Spring.
- La diversidad de formatos que pueden emplearse como contenidos de los cursos introduce la necesidad de contar con reproductores de dichos formatos. En una arquitectura cliente-servidor, con presentación basada en Web, una práctica común para solventar este problema es proporcionar este reproductor como Applet (o alguna tecnología similar como Flash) que use las librerías necesarias para el procesado del formato requerido y que serán descargadas bajo demanda. Esto aumenta el tiempo de iniciación del reproductor, y a menudo (sobre todo en usuarios que no gozan de una conexión con suficiente ancho de banda) infunde en el usuario desesperación, impaciencia y sensación de una herramienta de eLearning poco eficiente o mal diseñada. En una plataforma orientada a servicios, basada en el framework OSGi, estos reproductores residirían permanentemente en la máquina cliente, exportando su funcionalidad como servicios OSGi que serían usados cuando fuese requerido. Este reproductor puede aplicar la misma filosofía de OSGi, permitiendo la incorporación dinámica de futuros adaptadores para nuevos formatos.
- Otro problema radica en el propio concepto de SCO (Shareable Content Object) en SCORM. Está pensado para ser un contenido reutilizable, intercambiable. Esto es, un curso desarrollado una vez, puede ser reutilizado por diferentes centros, por diferentes LMS sin necesidad de modificación (si no se quiere personalizar, claro). Esto hace que el SCO no esté asociado para nada al LMS que lo "oferta". Sin embargo sí es necesario conocer dicho LMS en tiempo de ejecución para poder llevar a cabo el seguimiento del curso. Esto es, el propio SCO, a través del API de SCORM necesita comunicar con el LMS: para iniciar el curso, para indicar que se ha superado una unidad didáctica, para indicar cuándo el alumno abandona el curso momentáneamente y así poder contabilizar el LMS el tiempo invertido por el alumno hasta el momento, etc. Este enfoque es viable para sistemas basados en Web, dónde la interfaz gráfica del usuario consiste en una página generada dinámicamente por el propio LMS (por ejemplo mediante JSP, ASP, PHP, etc.) pero dificulta la implementación de un sistema autónomo, que ofrezca al alumno la posibilidad de seguir el curso sin estar conectado (off-line). Para ello sería necesario suplantar al LMS remoto por uno local encargado de guardar los eventos producidos para posteriormente, al volver al modo on-line, sincronizar con el LMS remoto enviándole estos eventos. Para ello es necesario poder asociar cada curso con el LMS remoto (en el servidor) encargado de llevar el seguimiento de dicho curso, pero no se dispondría de esta asociación.

3.4. Definición del problema

Se busca estudiar la adecuación de una arquitectura orientada a servicios, como base para el diseño de una plataforma versátil o universal que pueda ser útil para conseguir un eLearning más constructivista, y que permita ciertos niveles de personalización del aprendizaje, eliminando o por lo menos paliando, la mayoría de problemas listados en el punto anterior. Se desea poder instalar utilidades en tiempo de ejecución en la plataforma (clientes de chats, clientes de videoconferencia, juegos interactivos, etc.) sin necesidad de tener que reiniciar la plataforma y simular el comportamiento de un LMS de forma que puedan reproducirse cursos tanto con conexión a Internet como sin ella, por lo que será necesaria la distribución e instalación (que llamaremos despliegue a partir de ahora) de cursos (o de contenidos en general) en la plataforma, de forma que puedan reproducirse localmente sin conexión a ninguna red.

El primer problema que nos encontramos es cómo distribuir los contenidos. Siguiendo SCORM éstos deberían ser un archivo comprimido en formato ZIP, con un fichero descriptor según la especificación IMS CP. Éste formato es actualmente, ateniéndonos a la tabla comparativa de herramientas del capítulo anterior, el más aceptado. Será necesario encontrar un formato de distribución de contenidos que pueda ser aceptado tanto por plataformas de servicios OSGi como por la mayoría de LMS. Es un problema relacionado con el despliegue de contenidos.

El objetivo es conseguir una plataforma que permita entornos “pesados” con interfaces de usuario ricas en detalle, adecuadas por ejemplo para simuladores en tiempo real o entrenadores, sin renunciar a entornos web (clientes ligeros basados en tecnologías web: HTML, Applets, Flash, Shockwave...). También se desea poder interactuar con la plataforma usando controles remotos de simple manejo, como un mando a distancia de TV, para facilitar el desarrollo de entornos de eLearning con resolución adaptada a la TV digital o experiencias interactivas basadas en pantallas táctiles. Es problema relacionado con la integración de dispositivos hardware (y sus librerías nativas) y la gestión de eventos de bajo nivel (generación de eventos nativos de teclado y ratón principalmente). Por tanto se ha definir un conjunto de servicios que permitan llevar a cabo las tareas necesarias para la reproducción de los cursos, tanto on-line en comunicación con LMSs remotos como off-line, y definir los requisitos que deben cumplir las posibles aplicaciones que se instalen en esta plataforma y que estén destinadas a complementar el proceso de aprendizaje (chats, foros, videoconferencias, tablón de noticias, galerías de fotos, agendas, calendarios, herramientas de planificación, etc.)

Otro problema a resolver en la plataforma de servicios será el relativo a la interfaz gráfica. Será necesario abstraer las características comunes en la mayoría de interfaces gráficas para facilitar que pueda implementarse cualquier tipo de interfaz según requiera el dispositivo en que se encuentre, sin que ello afecte a los datos o la funcionalidad de los servicios registrados. Igualmente habrá que diferenciar qué aplicaciones cuentan con interfaz gráfica de usuario y cuáles no. También es deseable que todas las aplicaciones con interfaz gráfica de usuario, puedan clasificarse según su categoría (editores, reproductores, herramientas, juegos...) para facilitar el acceso desde una interfaz de usuario que haga las veces de escritorio (al estilo del escritorio de windows, o bien un escritorio web o página jsp, etc.) Este es un problema relacionado con la identificación de servicios en el que están involucradas tareas como la descripción del servicio, la clasificación en distintos tipos de servicios (con interfaz gráfica de usuario o sin ella) y la identificación en tiempo de instalación de estos y la categoría asociada a ellos, para que cuando correspondan puedan mostrarse clasificados por esta categoría. Es un problema relacionado con la taxonomía de aplicaciones y su procesamiento en función de cada tipo.

En cuanto a los problemas relacionados con la personalización del entorno y de los cursos, es necesario identificar cual puede ser el lugar idóneo para aplicar determinadas técnicas de adaptación, así como identificar estas técnicas. Son problemas relacionados con la personalización del entorno y la adaptabilidad de contenidos: técnicas de adaptabilidad.

Así pues identificamos en el trabajo diferentes áreas de actuación correspondientes con los siguientes problemas por resolver:

- Despliegue de componentes (contenidos, aplicaciones, librerías...)
- Integración de dispositivos y generación de eventos nativos de bajo nivel
- Taxonomía de aplicaciones.
- Técnicas de adaptabilidad

Por último un problema más relacionado con la implementación en particular de la herramienta de gestión del aprendizaje, puesto que es necesario definir un servicio OSGi que provea de la funcionalidad definida por el API de SCORM para la comunicación con el LMS remoto, y que puedan usar el resto de servicios registrados en la plataforma. Esta forma ya no sería conforme a la especificación de SCORM, pero permitiría que cursos no pensados para su visualización en un navegador (como por ejemplo simuladores, entrenadores en el manejo de instrumental técnico,

etc.) puedan igualmente ser compatibles con un LMS SCORM. Para los cursos SCORM no habría ningún problema, pues el API javascript puede incluirse en el bundle para ser usada en cursos pensados para reproducirse en un navegador web. El problema reside en el LMS local, que ha de decidir el entorno de ejecución para el curso entre el tradicional basado en un navegador web y API javascript, u otro basado en Swing, SWT, etc. Pensamos que la solución a este problema puede consistir en incluir una propiedad en el manifiesto del bundle (MANIFEST.MF) indicando el tipo de curso. Esta propiedad sería leída por el LMS local ante un evento de instalación o actualización del bundle, actuando en consecuencia. Por ejemplo, registrando la aplicación web usando el servicio HTTP de OSGi para un curso SCORM o usando un entorno gráfico a modo de escritorio y una aplicación de éste que haría las veces de reproductor del curso adecuado, tal y como se ha realizado en el proyecto Passepartout que se comenta en el siguiente punto de esta memoria.

El problema de la ausencia de asociación entre LMS y curso, puede resolverse al nivel de OSGi, sin afectar para nada a la conformidad de un curso con la especificación de paquete SCORM o IMS CP. Nos apoyaríamos para ello en el manifiesto del bundle. En la especificación de OSGi R4 se contempla la posibilidad de añadir propiedades al manifiesto. De esta forma podríamos definir una propiedad que pudiera utilizar el LMS local para poder sincronizar con el LMS remoto. Por ejemplo:

Bundle-RLMS: <http://www.telvent.com/LMS/LMSServlet>

Otra posibilidad podría consistir en almacenar la URL de descarga del curso, encargándose de ello la herramienta de despliegue, manteniendo así un registro de URLs de todos los bundles descargados, facilitando de esta forma la planificación de consultas periódicas para comprobar la existencia de nuevas versiones de bundles (hemos asumido que los paquetes IMS CP son cursos). Sin embargo, de esta forma no sería posible automatizar la actualización para bundles instalados manualmente por el usuario (por ejemplo desde otro PC del hogar, desde una memoria flash o tarjetas de memoria SD, desde un CD, etc.). Lo más recomendable, pues, es llevar a cabo las dos opciones pues no son incompatibles.

3.5.Propuesta de solución

Como primer paso, nos planteamos utilizar como base una arquitectura orientada a servicios, que nos facilite la integración de diferentes servicios, débilmente acoplados, que puedan interoperar entre sí. Esto repercutirá en una mayor escalabilidad y facilidad de mantenimiento de la plataforma. En cambio también es necesario, desde el punto de vista del despliegue de aplicaciones y contenidos en una plataforma cliente, un enfoque orientado a componentes, que nos permita tratar paquetes de contenidos (como por ejemplo los SCO de SCORM) como una unidad. Precisamente estos dos enfoques, Orientación a Servicios y Orientación a Componentes, son los utilizados por la arquitectura definida por la Alianza OSGi en su especificación OSGi (cuya versión actual es la R4), lo que motivó su elección para este trabajo.

Además, muchos de los problemas listados en el punto 3.3 están relacionados con la metodología de desarrollo del sistema. Por ello se pretende, al usar OSGi, aprovechar las ventajas que ofrecen entornos integrados de desarrollo (IDEs) como Eclipse (desarrollado y distribuido como un conjunto de bundles OSGi) a la hora de crear plataformas OSGi con interfaz de usuario rica (Eclipse Rich Client Platform) para crear herramientas para el desarrollo de cursos que puedan integrar el ciclo completo de vida de los mismos: desarrollo, pruebas y publicación.

Internamente la especificación OSGi proporciona un framework que permite gestionar servicios: añadir un servicio al registro, iniciar o finalizar un servicio registrado, eliminar un servicio del registro, así como consultar los diferentes servicios registrados (incluso utilizando filtros, que nos permitan refinar la consulta). La creación de servicios en el framework OSGi es sencilla y estructurada; existen muchos servicios obligados por la especificación que deben de ser tomados como base para la provisión de nuevos servicios. El framework OSGi también proporciona un mecanismo para solicitar una referencia a un servicio, de forma que otro servicio pueda usarlo, a

través del objeto `BundleContext`, que contiene el contexto asociado a un "bundle". Y es aquí donde confluyen ambos enfoques, pues si bien en tiempo de ejecución e internamente se corresponde claramente con una arquitectura orientada a servicios, si pensamos en el mecanismo de despliegue o instalación de estos servicios aparece el concepto de bundle: la mínima unidad de despliegue. Un bundle no es más que un archivo java (.JAR) que contiene todos los recursos y la lógica de la aplicación que se desea instalar en el framework de OSGi.

El bundle contiene un fichero, denominado Manifiesto, que describe mediante un conjunto de propiedades (nombre:valor) el contenido incluido en dicho bundle. Este conjunto de propiedades también está definido por OSGi, aunque los desarrolladores son libres de añadir nuevas propiedades según sus necesidades específicas si desean implementar nuevos frameworks OSGi. Un bundle puede desempeñar varias funciones: desde ser meramente una librería (un conjunto de clases que se exportan para que puedan ser utilizadas por otros bundles), a toda una aplicación de usuario (con interfaz gráfica de usuario, línea de comandos, etc). El API del framework de OSGi define la interfaz `BundleActivator`, que todo bundle que desee realizar alguna operación a partir del proceso de arranque debe implementar. Esta interfaz constituye, para el hilo de ejecución del framework, el punto de entrada en la lógica específica que se distribuye con el bundle, y define dos métodos, `start(BundleContext)` y `stop(BundleContext)`, que permitirán controlar las acciones a realizar para iniciar la lógica del bundle y para pararla. Mediante el método `start` el desarrollador instanciará los servicios (pueden ser ninguno, uno o más de uno) de la lógica de la aplicación contenida en el bundle y los registrará usando el contexto del bundle (que se pasa como parámetro), llevando a cabo la eliminación del registro y su destrucción en el código del método `stop`.

La funcionalidad para una aplicación concreta puede requerir de uno o más servicios OSGi repartidos en uno o más bundles, estableciendo de esta forma dependencias entre los bundles implicados.

OSGi especifica cómo gestionar el ciclo de vida: instalación, arranque, parada o actualización de un bundle, pero no tiene en cuenta cómo instalar el conjunto de bundles de los que éste depende, dejando esta cuestión a la implementación de la especificación del framework de OSGi.

Vamos a abordar uno a uno los principales problemas por resolver identificados en el punto anterior.

3.5.1.Despliegue de componentes

Se debe facilitar la utilización tanto on-line como off-line de los cursos permitiendo que el usuario pueda seguir los contenidos de un curso sin depender de la conexión a Internet, si previamente ha descargado el curso a la plataforma. Esto implica incluir cierta funcionalidad, tradicionalmente asociada al servidor en la plataforma cliente simulando el comportamiento del LMS para poder reproducir un curso. Podría implementarse o adaptarse un reproductor de cursos existente como `Reload SCORM Player`⁵³ que sería factible exportar como un servicio OSGi y distribuirse como un bundle. Esto traslada la lógica de reproducción del curso desde el lado del servidor al lado del cliente, y evidentemente, para cuando el usuario desee retomar el curso sin conexión a Internet, es necesario que dicho curso haya sido desplegado en la plataforma del cliente. Un punto a favor de OSGi es la compatibilidad de las unidades de despliegue, los bundles, con la especificación para empaquetados de contenidos más usada: IMS Content Packaging. Tanto IMS CP como OSGi usan el algoritmo de compresión ZIP, en el caso de OSGi en forma de archivos Java (.JAR). Dentro del paquete de contenidos ZIP, tanto OSGi como IMS CP definen un fichero descriptor del contenido: el Manifiesto, pero a pesar de ser léxica, sintáctica y semánticamente diferentes (el manifiesto de OSGi hace uso del fichero `MANIFEST.MF` de la especificación de archivos Java y se basa en pares propiedad-valor, mientras que el manifiesto de IMS CP está escrito en XML) no son mutuamente excluyentes en un mismo paquete, pues su ubicación y nominación

⁵³ <http://www.reload.ac.uk/scormplayer.html>

son diferentes, al ubicarse el de IMS CP en la raíz (parte más externa) del paquete ZIP y tener por nombre `imsmanifest.xml`.

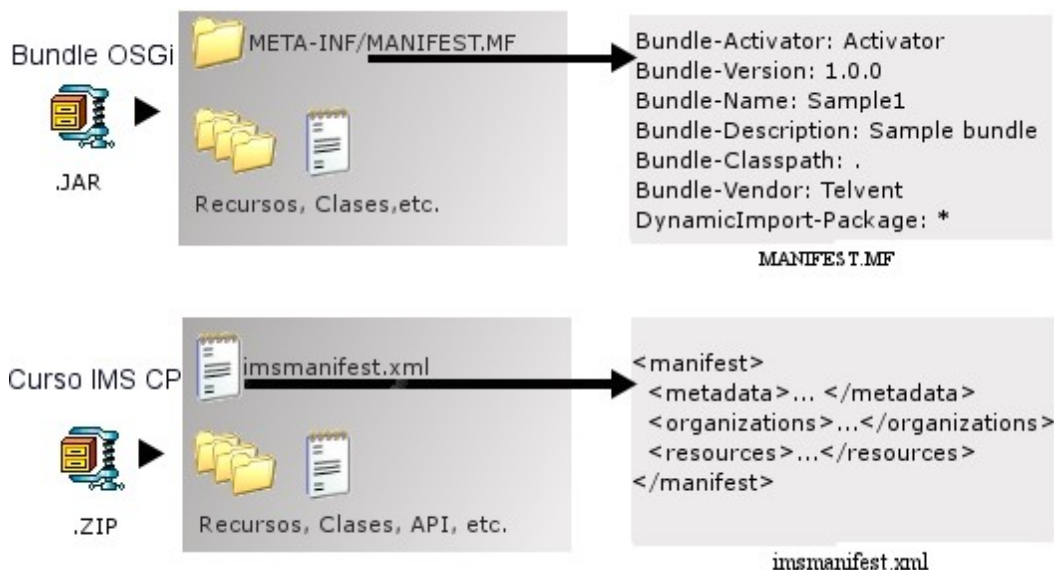


Figura 17. Semejanzas en el empaquetado de cursos y bundles OSGi

Esto nos lleva inmediatamente a pensar: "¿por qué no distribuir los cursos empaquetados con IMS CP (sean o no SCORM) como bundles de una plataforma OSGi?" Para el framework OSGi, el manifiesto de IMS CP no dejaría de ser un recurso más. En cambio es posible implementar un servicio escuchador, que implemente la interfaz `org.osgi.framework.BundleListener` y escuche los eventos relacionados con el despliegue de bundles, tal y como define la especificación de OSGi. Este servicio detectará las instalaciones, desinstalaciones de los bundles así como la actualización de estos, y podrá detectar en tiempo de ejecución si el bundle se trata de un curso o no (simplemente detectando si el paquete contiene el fichero `imsmanifest.xml` en el directorio raíz), pudiendo procesar el curso para llevar a cabo las acciones convenientes (catalogación, notificación, etc)

Al tratar los SCOs como bundles OSGi solventamos el problema de la distribución de éstos a plataformas OSGi, pero tal y como se describe en [35] existen otros problemas relacionados con el despliegue de componentes y las dependencias entre estos.

En OSGi existen problemas parecidos a los existentes a nivel de sistema operativo. Las unidades mínimas de despliegue (los bundles) pueden no formar una aplicación por sí solos, pudiendo ser necesario un conjunto de ellos (dependientes entre sí) para proveer la funcionalidad completa de la aplicación. Esto obviamente depende de la implementación seguida en el diseño de los bundles. Aparece en este punto el concepto de "aplicación de usuario final". Esta aplicación ofrece una cierta funcionalidad final al usuario, que puede interactuar con la aplicación mediante alguna interfaz de usuario (gráfica, texto, con dispositivos de entrada, etc) y estará formada por al menos un bundle iniciador, pudiendo además ser necesaria el despliegue e iniciación previa de uno o más bundles de los que depende funcionalmente (porque use servicios registrados por éstos). En versiones anteriores de OSGi esto supuso un problema pues sólo se controlaban las dependencias relacionadas con la visibilidad de clases, recursos, etc. Pero no dependencias funcionales. Ilustramos el problema a continuación.

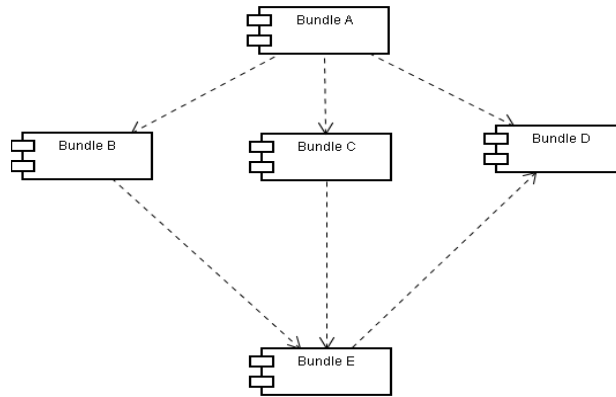


Figura 18. Grafo de dependencias a nivel de paquetes

El grafo de la figura anterior muestra las dependencias a nivel de paquetes entre cinco bundles: A depende para su funcionamiento de los bundles B, C y D que deben haber sido instalados previamente al A. A su vez tanto B como C dependen de E, que depende de D. Esto hace que este conjunto de bundles deba de ser instalado siguiendo alguna de las siguientes secuencias:

- D-E-B-C-A
- D-E-C-B-A

Estas dependencias son a nivel de clases. Es decir, que para que B pueda ser instalado correctamente, necesita que haya sido instalado previamente E, puesto que en alguna clase del bundle B se importa alguna clase incluida en el bundle E. Estas dependencias pueden deducirse a partir de los Manifiestos de cada bundle, según la lista de paquetes importados y exportados:

```
Bundle-ManifestVersion: 2
Bundle-Activator: org.gateway.home.desktop.DesktopBundleActivator
Bundle-ContactAddress: http://www.telvent.com
Bundle-Name: Desktop
Bundle-SymbolicName: Desktop
Bundle-Version: 2.0.6
Bundle-Description: Swing-based Desktop
Bundle-DocURL: http://forge.os4os.org/projects/desktop
Bundle-Vendor: Telvent
Bundle-Category: service
Import-Package: org.osgi.framework,
org.gateway.home.desktopbase.utils,
org.gateway.home.desktopbase.service,
org.gateway.home.desktopbase.components,
org.ungoverned.osgi.service.shell,info.clearthought.layout,
org.gateway.home.desktopbase.applications.shellgui,
org.ungoverned.osgi.service.bundlerepository,
org.osgi.service.log
Export-Package: org.gateway.home.desktop,
org.gateway.home.desktop.components,
resources.images
```

Figura 19. Manifiesto de un bundle.

En OSGi, cada bundle tiene su propio cargador de clase y su propio espacio de clases, por lo que al menos que se indiquen las clases que se desean incorporar a dicho espacio, las clases exportadas por otros bundles no serán "visibles". Es el framework el que se encarga de referenciar entre los espacios de clases de cada bundle, para que clases exportadas por un bundle puedan ser visibles desde el espacio de clases de otro que la importa. En la figura anterior se muestra el fichero Manifiesto de un bundle. En él se especifican los paquetes necesarios que se deben añadir al classpath del bundle, mediante la propiedad `Import-Package` y los que este bundle exporta para que otros bundles puedan usarlas, mediante la propiedad `Export-Package`. Como se puede apreciar, todo se hace a nivel de paquetes, e incluso los recursos (imágenes, vídeos, etc) pueden exportarse e importarse indicando en alguna de las propiedades descritas el paquete en el que se encuentran. Sin embargo a partir del Manifiesto no se indica en ningún momento la dependencia con otros componentes, es decir, con otros bundles. Esa lista de importación de paquetes puede dar lugar a dependencias con 1, 2...n bundles (dónde n coincide con el número total de paquetes requeridos en la propiedad `Import-Package`) pero no hay forma de saber cuántos ni qué bundles nos resuelven estas dependencias. Incluso podría suceder que pudieran encontrarse varias soluciones (debido a malas prácticas como la duplicación de paquetes o librerías en más de un bundle) ante lo cual sería conveniente decidir una política de despliegue a seguir: desplegar la solución con menor número de bundles, la que contenga bundles de menor tamaño, etc.

Más complicado es a partir de los propios bundles, identificar las dependencias funcionales. Supongamos el diagrama de la siguiente figura:

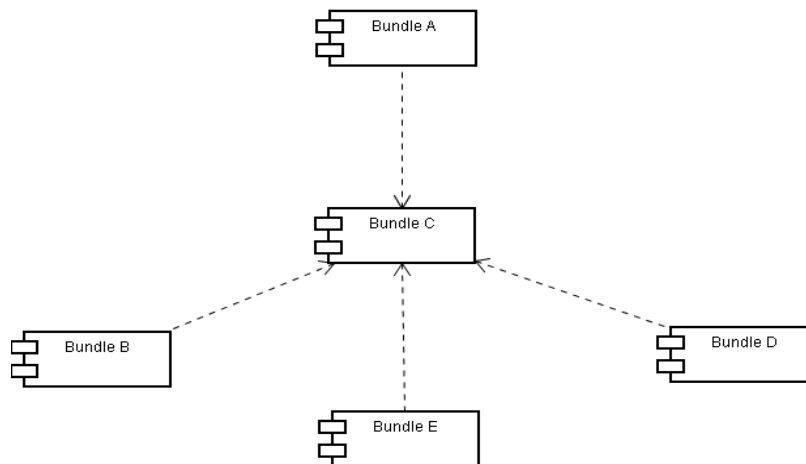


Figura 20. Grafo de dependencias funcionales entre bundles

Supongamos que el Bundle C registra un servicio encargado de recoger datos proporcionados por otros bundles (en este caso B, D y E) y almacenarlos de forma persistente. Por su parte el Bundle A utiliza otro servicio del Bundle C que proporciona un conjunto de datos en respuesta a una consulta recibida (bien con filtros, con un lenguaje de consultas, etc.) y los presenta al usuario final con una interfaz gráfica de usuario atractiva, de forma que el usuario pueda solicitar los datos necesarios interactuando a través de esta interfaz gráfica. Supongamos que la representación de los datos no es el problema (podemos imaginar un bundle que exporte los paquetes que definen el diccionario de datos posibles a usar, o que se devuelven en XML y es conocido el DTD o Schema que especifica la sintaxis del documento, etc.) Según el diagrama anterior, el Bundle A sólo depende del Bundle C, pues necesita obtener el servicio o servicios de C que utilizará. Sin embargo, puede ser que la interfaz gráfica esté diseñada para presentar datos ofrecidos por los Bundle B, D y E, o que a través de algún servicio OSGi de C se realicen operaciones sobre algunos de estos datos previamente proporcionados por alguno de estos tres bundles. Si estos tres bundles no están disponibles, la funcionalidad ofrecida por A no estará completa. Y aunque en cuanto a importación y exportación de paquetes Java, o de recursos, no

existen dependencias, sí que existen dependencias funcionales en tiempo de ejecución. De ahí la importancia de definir el concepto de “aplicación final de usuario” desde un punto de vista de la funcionalidad ofrecida e indicar de alguna forma los bundles implicados, los bundles que componen dicha “aplicación final de usuario”. En este caso concreto, no importa el orden en que hayan sido instalados los bundles, sino que depende del proceso de ejecución, de las consultas realizadas por el usuario. Son problemas en tiempo de ejecución y no en tiempo de despliegue.

Asociado a este problema de la gestión del despliegue, también nos encontramos con que la pista necesaria para comenzar la resolución de las dependencias de un bundle se encuentra precisamente en el Manifiesto, esto es, dentro del propio bundle, por lo que en el proceso de resolución de dependencias sería necesario descargar previamente cada bundle desde su ubicación antes de poder procesar las dependencias que éste pueda tener a su vez. ¿Y si en mitad del proceso se produce un error: se cae la red, el formato no es el adecuado, etc? La penalización sería más alta cuanto más bundles se llevasen descargados y cuanto mayor fueran en tamaño éstos.

La solución propuesta pasa por delegar en un Centro de Control, una entidad que gestionaría las peticiones de varias plataformas OSGi, una especie de servidor encargado de proporcionar información que facilite a la plataforma la resolución de las dependencias, y facilitando los bundles desde un *repositorio* o almacén de bundles.

La arquitectura del lado servidor, llamado Centro de Control, está basada en un servidor de aplicaciones J2EE[47]; la arquitectura del lado cliente está basada en OSGi; para comunicar ambos extremos se ha escogido usar Servicios Web (SOAP), una tecnología neutral que asegura la compatibilidad ante posible cambios futuros o la posible integración con plataformas de distinta base tecnológica.

Centro de Control

En el Centro de Control se sitúan las funciones de soporte necesarias para hacer provisión de servicios. En realidad los únicos ficheros que se van a desplegar son los bundles, pero son necesarios además descriptores que aporten la información de los bundles y las aplicaciones. En dichos descriptores se indica información descriptiva sobre los bundles y aplicaciones e información relativa a las dependencias que han de satisfacerse para su despliegue. El motivo de tener separados los descriptores de los propios bundles es que así se evita tener que obtener el bundle completo (que es un fichero comprimido JAR, es decir, un ZIP) para el análisis de las dependencias, y sólo es necesario obtener un pequeño fichero de texto con la información sobre estas dependencias. Una vez decidido el conjunto de bundles a instalar podremos desplegar únicamente estos sobre la plataforma.

Además del repositorio necesitamos una lógica de negocio para gestionar estos repositorios. El Servidor del Catalogo de Servicios [36], SCServer, ayuda a navegar por las aplicaciones disponibles y escoger qué aplicación del catálogo instalar.

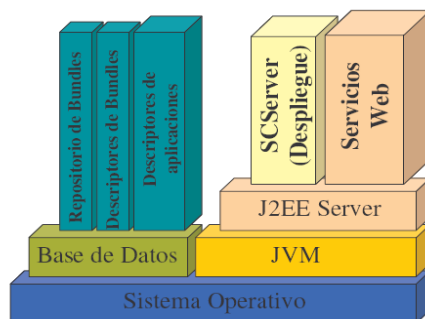


Figura 21. Arquitectura del Centro de Control

Las comunicaciones están soportadas mediante servicios web, basados en la implementación de Apache de SOAP, Axis [48], que incluye un motor de gestión de servicios web.

Plataforma OSGi

Las herramientas que conforman el sistema de despliegue en la plataforma OSGi son:

- J-Bones [40], encargado de la resolución de dependencias a nivel de bundle;
- OAM [49], encargado de la gestión de la información y ciclo de vida de la aplicación;
- El cliente del Catálogo de Servicios, SCClient, [36] que se encarga de consultar y operar sobre el repositorio de aplicaciones comunicándose con su homólogo en el lado del servidor, SCServer, a través de servicios web. Para soportar estos servicios web ha sido necesario desarrollar un bundle basado en Axis, que proporciona un servicio OSGi que envuelve la funcionalidad proporcionada por Apache Axis [50].
- También es necesaria APT-GET [34], una herramienta del sistema de gestión de paquetes a nivel de sistema operativo en el que se apoya J-Bones para instalar dependencias de bundles hacia paquetes nativos.

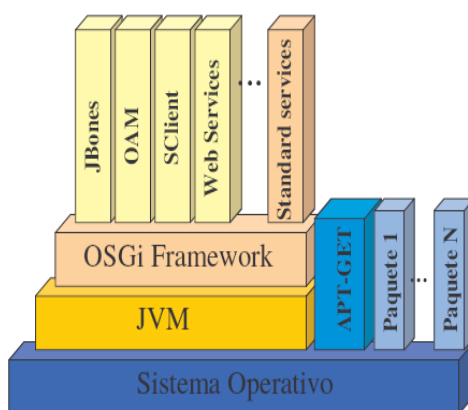


Figura 22. Arquitectura de la plataforma de servicios OSGi

Estas herramientas han sido desarrolladas en el marco del proyecto europeo ITEA-OSMOSE, que se describe más adelante en el capítulo 4.

Por tanto hemos definido dos tipos de dependencias, funcionales o a nivel de aplicación, y de paquetes o a nivel de bundles.

Modelo de aplicación y descriptores de despliegue

Una aplicación final de usuario se define como dependencias hacia bundles y otras aplicaciones, siendo OAM el encargado de procesar estas últimas y quedando fuera de su dominio el analizar o conservar información sobre las dependencias entre bundles, de las que se encargará la herramienta J-Bones. Ésta última no analiza ni guarda información sobre las dependencias a nivel de paquetes nativos, cuya responsabilidad recaerá en APT-GET. Este modelo se muestra en la siguiente figura.

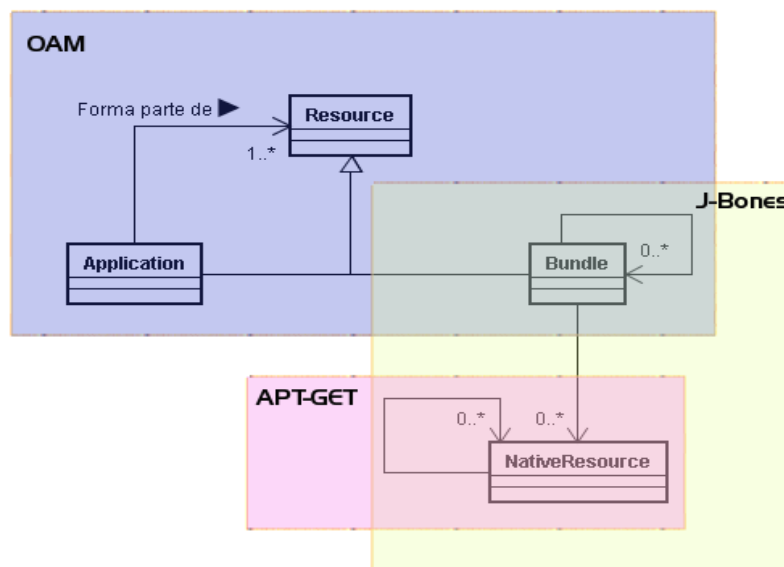


Figura 23. Modelo de aplicaciones

Como hemos definido dos ámbitos de definición de dependencias, a nivel funcional y a nivel de paquetes, y la aplicación de dos herramientas encargadas de la resolución de éstas (más una tercera para la instalación de librerías nativas en Linux) mostramos a continuación los ficheros descriptores tanto de las aplicaciones como de los bundles con los que ambas herramientas trabajan:

Descriptor de J-Bones

En este descriptor XML hay dos bloques fundamentales, la etiqueta `<source>` y la etiqueta `<endPoint>`. En la primera etiqueta se muestra la información del bundle que representa este descriptor y por razones obvias sólo puede haber una de estas etiquetas en cada descriptor; la segunda etiqueta define una dependencia del bundle, con otro bundle o con un paquete del sistema operativo, por lo que puede haber cualquier número de estas etiquetas (o ninguna).

Dentro de la etiqueta `<source>` destacan otras dos:

- `<priority>` indica la importancia del bundle
- `<container>` representa el fichero que contiene el bundle

Dentro de la etiqueta `<endpoint>` destacan otras tres:

- `<level>` indica si la dependencia es a nivel OSGi o a nivel nativo
- `<kind>` indica el tipo de dependencia: AND, NOT, OR o XOR.
- `<container>` representa el fichero que contiene esta dependencia

La etiqueta `<service>` se puede encontrar dentro de los dos bloques principales. En ambos casos indica si un bundle proporciona un servicio (o servicios, puede haber más de una etiqueta) determinado, lo cual puede evitar el despliegue de servicios repetidos en la plataforma.

Descriptor OAM

El descriptor de aplicación de OAM consta de dos bloques fundamentales que se incluyen dentro de la etiqueta <application>:

- El contenido de la etiqueta <information> está dedicado a describir la aplicación. Esta información es útil tanto para ser procesada por el cliente a la hora de decidir o no su instalación en función de las características de la plataforma, como para mostrar al usuario. Incluye el nombre de la aplicación y su versión; nombre, descripción y URL del proveedor de la aplicación; información útil para mostrar al usuario; icono identificativo de la aplicación; autor de la aplicación y categoría a la que pertenece (ej. juegos, sistema, música, utilidades...)

- La etiqueta <resources> contiene todas las dependencias de la aplicación.

Pueden expresarse tres clases de dependencias: hacia aplicaciones, hacia bundles concretos y hacia descriptores JBones. Cada clase de dependencia puede estar presente o no, y puede haber un número ilimitado de cada tipo. Si el orden de instalación de estas dependencias es importante, se puede especificar mediante la etiqueta <order> presente en todos los tipos de recursos.

A continuación se muestra un ejemplo de descriptor de OAM:

```
<application>
  <information>
    <name>Hogar Digital</name>
    <version-info>
      <number>1.0</number>
      <release>1</release>
    </version-info>
    <vendor-info>
      <vendor-name>TelefónicaI+D</vendor-name>
      <vendor-description>R&D company</vendor-description>
      <vendor-url>http://www.tid.es</vendor-url>
    </vendor-info>
    <user-info>
      <display-name> Hogar Digital </display-name>
      <description>Servicio de gestión del Hogar Digital</description>
      <homepage>http://www.tid.es/repo/img.gif</homepage>
    </user-info>
    <icon>
      <url>http://www.tid.es/repo/img/imagen1.gif</url>
    </icon>
    <author>M. García</author>
    <category>
      <id>HA</id>
      <name>Home Automation</name>
      <description>Applications related with Home Automation</description>
    </category>
  </information>

  <resources>
    <descriptors>
      <descriptor>
        <type>jbones</type>
        <descriptor-url>
          http://www.tid.es/repo/descriptors/hogar_digital.xml
        </descriptor-url>
        <order>3</order>
      </descriptor>
    </descriptors>
```

```
<bundles>
  <bundle>
    <location>http://www.tid.es/repo/bundles/cm.jar</location>
    <order>2</order>
  </bundle>
</bundles>
<app-dependencies>
  <app-dependency>
    <name>X10 Support</name>
    <version-info>
      <number>1.0</number>
      <release>1</release>
    </version-info>
    <descriptor-url>
      http://www.tid.es/repo/applications/x10-1.1.0.xml
    </descriptor-url>
    <order>1</order>
  </app-dependency>
</app-dependencies>
</resources>
</application>
```

Listado 2. Ejemplo de fichero descriptor de OAM

El proceso de despliegue

Cuando se crea una aplicación se debe ser cuidadoso con lo que se asocia a esta. La instalación de las dependencias es suficiente para el funcionamiento, pero no debemos asociar a la aplicación solamente las nuevas dependencias instaladas, sino también las que ya estaban instaladas en la plataforma.

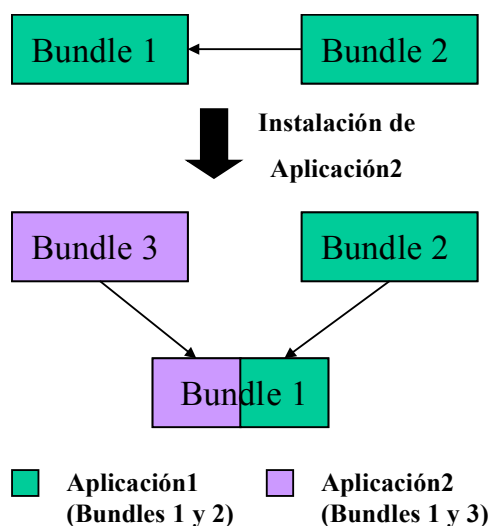


Figura 24. Asociación de bundles y aplicaciones

- Aplicación 1 depende de los bundles Bundle 1 y Bundle 2.
- Aplicación 2 depende de los bundles Bundle 1 y Bundle 3.

Si instalamos Aplicación1 se instalarán los bundles 1 y 2 pues son nuevos en la plataforma. Al instalar Aplicación2, si sólo asociásemos los nuevos bundles instalados estaría compuesta únicamente por Bundle3, se podría crear una inconsistencia en el futuro si se para o desinstala Aplicación1. Para evitarlo hay que asociar a la aplicación no sólo los nuevos recursos instalados sino todos los que se encuentran en el árbol de dependencias resultado del análisis. De esta forma, al instalar Aplicación2 el resultado del análisis de las dependencias debe dar como resultado que se asocian los bundles 1 y 3 a esta aplicación. En este caso, parar o desinstalar Aplicación1 no afecta a Aplicación2.

Pasos para instalar una aplicación:

1. Un usuario del sistema consulta las aplicaciones disponibles. Esto puede llevarse a cabo usando el cliente del Catálogo de Servicios [36] que de forma transparente al usuario comunica con el método `getAvailableServices()` de su correspondiente servicio en el lado del servidor (SCServer).
2. Como respuesta el método anterior devuelve una lista de URLs, cada una apuntando a un descriptor de aplicación.
3. Ya en la pasarela, se procesa cada descriptor para mostrar en pantalla la información asociada a cada aplicación.
4. El usuario decide instalar una aplicación. OAM empieza a procesar las dependencias funcionales especificadas en el descriptor correspondiente. Si la dependencia es una aplicación, se lanza de forma iterativa el proceso de instalación para esta dependencia; si es un bundle se instala desde la URL especificada; y si es un descriptor JBones se lanza un proceso de resolución de dependencias a nivel de paquetes.
5. Cuando acaba el análisis e instalación de dependencias de un descriptor JBones, se asocian todos los recursos del árbol solución.

Si el proceso de instalación de las dependencias falla (no se encuentra un bundle, un análisis de dependencias no llega a ninguna solución válida, etc.) se deberán desinstalar todas las dependencias instaladas hasta ese momento para dejar la plataforma en el mismo estado en el que se encontraba al comienzo.

3.5.2. Integración de dispositivos y generación de eventos nativos de bajo nivel

La propia arquitectura orientada a servicios facilita la integración e interoperatividad entre servicios. El problema de la integración de dispositivos se reduce al diseño e implementación de servicios capaces comunicar con dichos dispositivos. La integración de dispositivos, como por ejemplo un control remoto, puede implicar la existencia de dependencias con librerías nativas, como LIRC⁵⁴ o WinLIRC⁵⁵. Si existe algún API Java que facilite la comunicación con la librería nativa, como en este caso `jLirc`⁵⁶, deberemos crear un bundle que a partir de este API nos proporcione la comunicación deseada con el dispositivo y nos facilite el acceso a la funcionalidad nativa.

Este fue un problema abordado en el proyecto *Passepartout*, y que se comenta en el siguiente capítulo, ofreciéndose un resumen de como se consiguió la integración de un control remoto por infrarrojos para controlar cualquier aplicación gráfica.

3.5.3. Taxonomía de las aplicaciones

En ocasiones es necesario identificar el tipo de aplicación a desplegar en la plataforma, y hacerlo en tiempo de instalación, para poder procesarla adecuadamente. Por ejemplo, en una plataforma que cuenta con una interfaz gráfica a modo de escritorio, será interesante saber si una aplicación cuenta con interfaz gráfica de usuario y cómo podrían integrarse todas las aplicaciones con interfaz gráfica (por ejemplo ventanas Swing) con el gestor de ventanas del escritorio, puesto

⁵⁴ Linux Infrared Remote Control <http://www.lirc.org>

⁵⁵ Windows Infrared Remote Control <http://winlirc.sourceforge.net/>

⁵⁶ Java API for LIRC and WinLIRC <http://jlirc.sourceforge.net/>

que aquellas que no cuenten con interfaz gráfica de usuario no afectarán al escritorio (no añadirá un botón lanzador de la aplicación ni un enlace en el tapiz del escritorio, etc). Por tanto ya se establece a este nivel una primera clasificación de aplicaciones entre aquellas que cuentan con interfaz gráfica y las que no. Por otra parte, las primeras pueden especializarse aún más según diferentes categorías predefinidas por el escritorio: Accesorios, Aplicaciones, Juegos, Cursos, etc. y dentro de éstas otras categorías definidas por la propia aplicación (generalmente coincidiendo con el propio nombre de la aplicación).

Para solucionar este problema se opta por la definición de un sencillo API que permita la implementación de diferentes "escritorios" gestores de aplicaciones con interfaz gráfica de usuario, ya sea basadas en web (HTML + Servlets usando el HTTP Service de OSGi), Swing, SWT, o cualquier otra tecnología. En primer lugar definimos la funcionalidad que cualquier aplicación final de usuario ha de tener mediante la interfaz Application. Posteriormente ofrecemos dos especializaciones: una aplicación cliente (Swing, SWT, etc) a la que denominamos GUIApplication y una aplicación con interfaz gráfica basada en web a la que llamamos WebApplication, tal y como se puede apreciar en la siguiente figura:

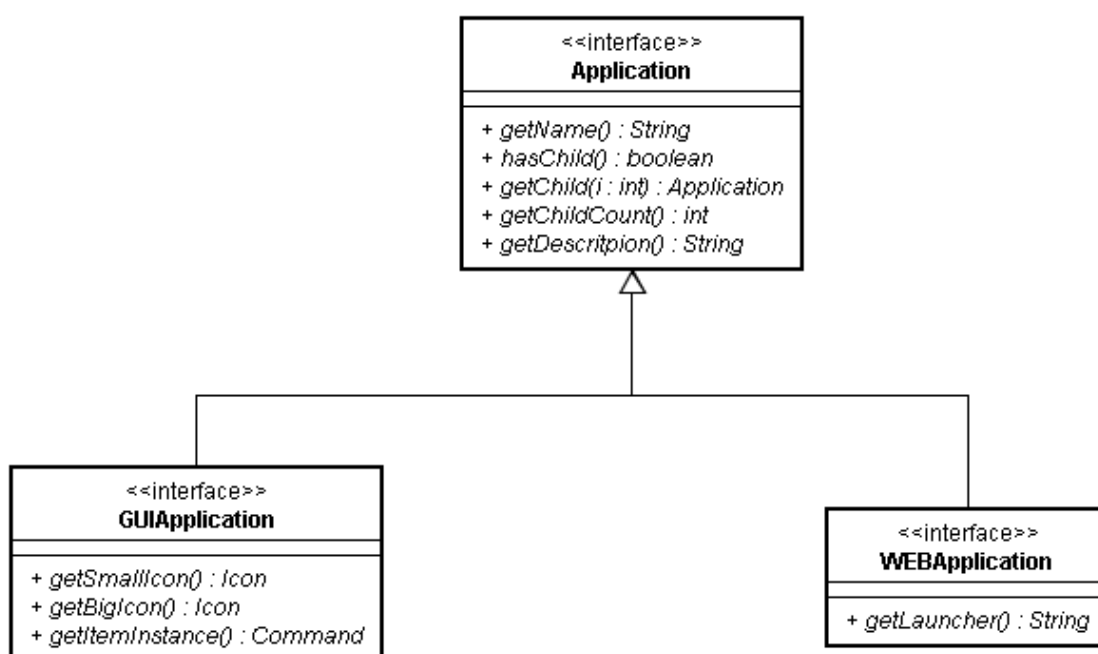


Figura 25. Modelo de aplicaciones con interfaz gráfica de usuarios

Este modelo está incluido en un bundle denominado `desktopbase` que define las características básicas de aplicaciones con interfaces gráficas de usuario. En el diagrama anterior, podemos observar cómo cualquier aplicación se caracteriza por un nombre, una descripción, y puede tener una serie de aplicaciones hijas (tantas como devuelva el método `getChildCount`). Esto permite el anidamiento de aplicaciones, y la clasificación de aplicaciones en menús jerárquicos.

Para las aplicaciones basadas en ventanas, además es necesario proporcionar un icono (en dos versiones: pequeña, pensado para la barra de título; y grande, pensado para un posible enlace de escritorio) y el comando que servirá de lanzador de la aplicación. Este comando puede hacer las veces de botón del menú por ejemplo, que al pulsarlo inicia la aplicación. Como se puede apreciar el inicio de una aplicación no está asociado al inicio del bundle. El ciclo de vida de una aplicación puede ser más corto, e incluso pueden iniciarse varias instancias de una aplicación (tantas como

veces ejecutemos el Comando que la inicia, dependiendo si la lógica de la aplicación así lo permite). Sin embargo la parada de un bundle o su desinstalación sí determina el fin de todas las instancias de aplicaciones activas de ese bundle. Para implementar este punto de vista, se optó por implementar las aplicaciones finales de usuario como servicios OSGi, registrándolas como servicios OSGi en el framework, mediante alguna de las interfaces *Application* disponibles. Esta gestión de las aplicaciones ha de hacerlas por tanto un servicio OSGi y no el framework, pues ocurre en el proceso de arranque del bundle. Para ello se define el servicio *GUIService*, que gestiona las aplicaciones con interfaz gráfica. El servicio *GUIService* implementa la interfaz *ServiceListener* del API del framework OSGi, que le permite recibir eventos del framework notificando los cambios de estados relacionados con los servicios OSGi: nuevo servicio registrado, servicio eliminado del registro de servicios OSGi, servicio actualizado, etc...

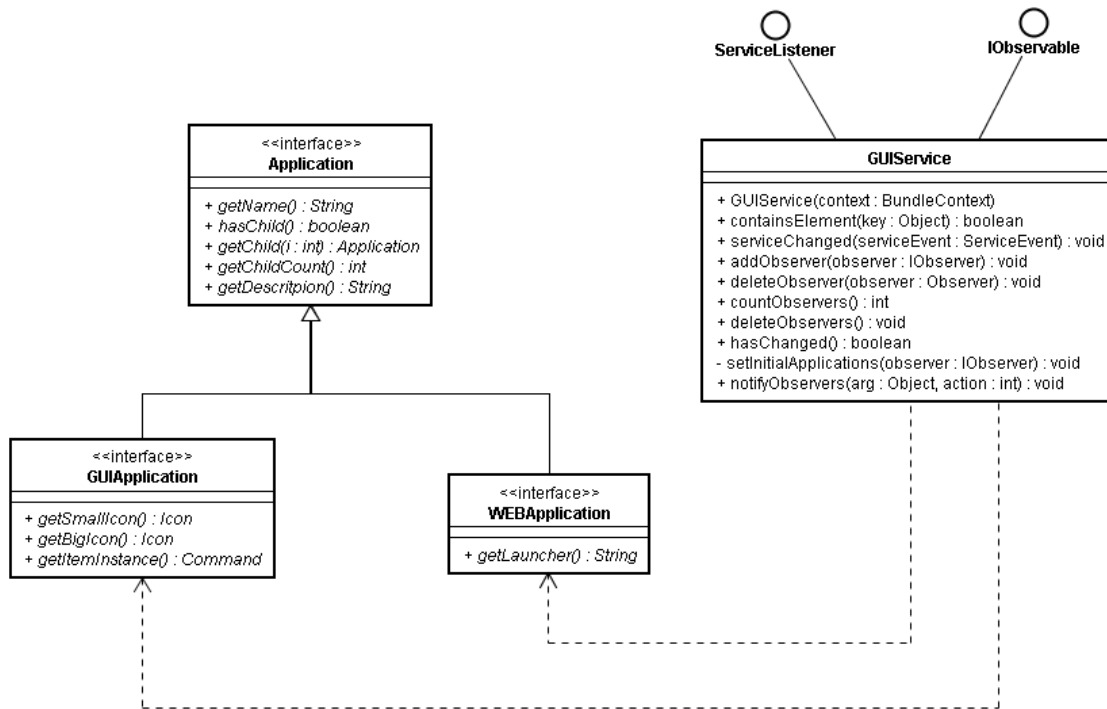


Figura 26. Gestión de aplicaciones finales de usuario por GUIService

Se ha aplicado el Patrón Observador para detectar en tiempo de ejecución las aplicaciones instaladas en el framework OSGi. El servicio *GUIService* será el objeto *observable* que notificará un cambio en su conjunto de aplicaciones instaladas a todos los observadores (todos los escritorios que pudiera haber). Estos observadores serán los distintos gestores de ventanas instalados en la plataforma. Es decir, *GUIService* se encarga de la identificación y diferenciación de diferentes aplicaciones gráficas, que notificará a todos los gestores, quiénes, comprobando el subtipo de aplicación gráfica decidirán si la procesan o no. Esto nos abre las puertas para poder soportar más de un escritorio (uno para plataforma pc, otro para plataforma PDAs y teléfonos móviles, etc) e incluso varios escritorios simultáneos (uno basado en ventana, otro basado en web y accesible remotamente mediante HTTP, etc.)

Para facilitar la implementación de estos gestores de aplicaciones gráficas, se definen dos interfaces: *DesktopInterface*, que define la funcionalidad que deben implementar todos los "escritorios" de aplicaciones gráficas (instancias de *Application*), y *DesktopManagerInterface*, donde se define la funcionalidad relacionada con la gestión de estas aplicaciones.

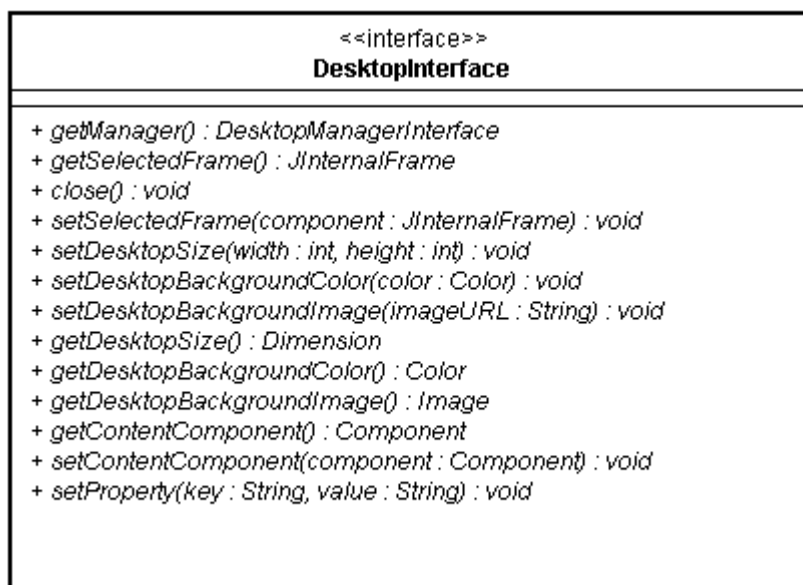


Figura 27. Interfaz DesktopInterface

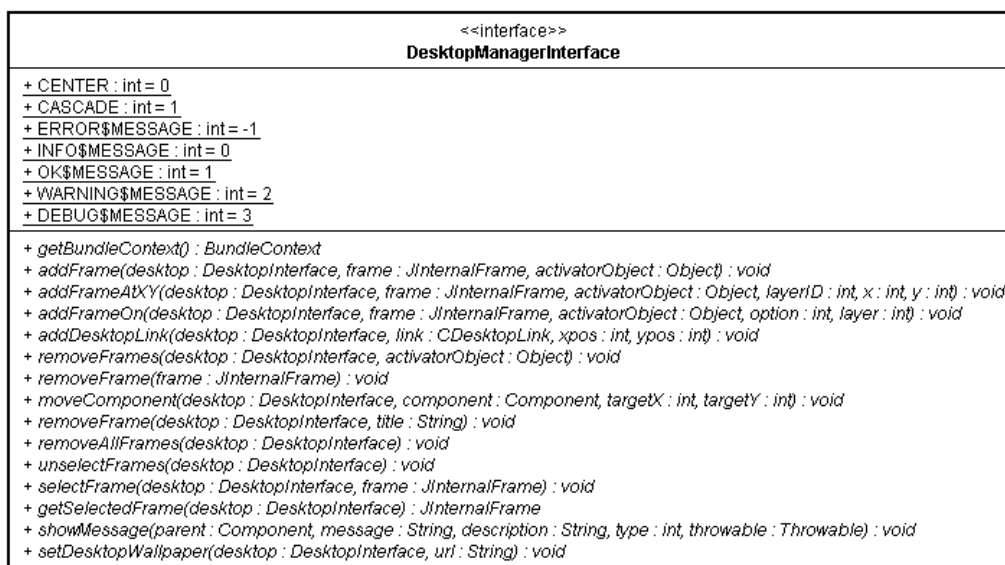


Figura 28. Interfaz DesktopManagerInterface

El funcionamiento es sencillo, cuando un bundle quiera instalar una aplicación gráfica en el escritorio deberá registrar en el framework un servicio que implemente `GUIApplication` o `WEBApplication`. El servicio `GUIService`, al implementar `ServiceListener` se percatará de este registro y añadirá automáticamente el comando al menú de aplicaciones mediante la instanciación de la clase cuyo nombre se pasa a través del método `getItemClassName()` de la

interfaz `GUIApplication` o añadirá un hiper-enlace a la página HTML que hace de escritorio cuyo texto sea el nombre de la aplicación y URL del hiper-enlace la que devuelve el método `getLauncher()` en el caso de ser una aplicación instancia de `WEBApplication`.

Para concretar este enfoque se construyeron dos ejemplos de escritorios: uno basado Swing a partir de la interfaz gráfica del proyecto EDUKA (que se comenta en el capítulo 4) y otra basada en OpenLaszlo⁵⁷ para interfaces web.

Este enfoque permitió, a los desarrolladores la integración de forma muy fácil de aplicaciones con interfaz gráfica de usuario en el escritorio. A continuación se muestra una breve guía para el desarrollo de aplicaciones finales de usuario con aplicación gráfica para el escritorio (basado en Swing)

Creación de aplicaciones *Desktop-Friendly*

Vamos a hacer como ejemplo la aplicación reloj, usando unas librerías de IBM para mostrar un reloj en pantalla, que toma la hora del sistema y puede mostrarla tanto en formato analógico como en formato digital.

Pasos

1. **Crear el activador del bundle.** Es necesario para poder instalar y arrancar el bundle en el framework OSGi. En este activador, en el método `start`, registraremos todas las *aplicaciones gráficas* que se incluyan en el bundle y que se quieren integrar en el Desktop.

Ejemplo:

```
package org.os4os.dskclock;

import org.osgi.framework.BundleActivator;
import org.osgi.framework.BundleContext;
import org.osgi.framework.ServiceRegistration;

public class Activador implements BundleActivator {
    ClockApp apli = null;
    ServiceRegistration reg = null;

    public Activador() {
        apli = new ClockApp();
    }
    public void start(BundleContext bundleContext) {
        reg = bundleContext.registerService("org.os4os.dskclock.ClockApp",
        apli, null);
    }

    public void stop(BundleContext bundleContext) {
        bundleContext.ungetService(reg.getReference());
        reg.unregister();
    }
}
```

En este ejemplo `ClockApp` debe implementar `GUIApplication`.

⁵⁷ OpenLaszlo, <http://www.openlaszlo.org/>

2. Implementación de la aplicación gráfica.

Implementar GUIApplication. (En realidad GUIService tienen en cuenta cualquier aplicación -interfaz Application- tenga o no interfaz gráfica, pero para el escritorio lo que nos interesa es GUIApplication). GUIApplication hereda de Application, por tanto debemos implementar tanto los métodos de una como de otra interfaz, que al final son los que podemos ver en el siguiente ejemplo:

```
package org.os4os.dskclock;

import javax.swing.Icon;
import javax.swing.ImageIcon;

import org.gateway.home.desktopbase.components.*;
import org.gateway.home.desktopbase.service.GUIApplication;

public class ClockApp implements GUIApplication {

    Launcher boton = null;

    public ClockApp() {
        boton = new Launcher();
    }

    /**
     * Devuelve la aplicación i-ésima
     * @param i int
     * @return Application
     */
    public GUIApplication getChild(int int0) {
        return null;
    }

    /**
     * Devuelve el número de GUIApplications hijas. En este caso 0. Si tuviera
     * una o más de una, haría que se anidasen en el menú del escritorio.
     * @return int
     */
    public int getChildCount() {
        return 0;
    }

    /**
     * Devuelve la instancia que implementa la interfaz Command, y que lanza la
     * aplicación.
     * getItemClassName method
     * @return
     */
    public Command getItemInstance () {
        return this.boton;
    }

    /**
     * Devuelve una descripción informativa de la aplicación
     * @return String
     */
    public String getDescriptpion() {
        return "Analog and digital clock";
    }

    /**
```

```
* Devuelve el icono representativo de la aplicación
*/
public Icon getIcon() {
    return getBigIcon();
}

/**
 * Devuelve el nombre de la aplicación
 * @return String
 */
public String getName() {
    return "The Clock";
}

/**
 * Devuelve true si la aplicación contiene una o más aplicaciones hijas.
 * Este caso aparece en sub-menus, aplicaciones con más de un punto de
 * entrada o formas de arranque, etc.
 * @return boolean
 */
public boolean hasChild() {
    return false;
}

/**
 * Devuelve el Icono pequeño asociado a la aplicación
 * @return Icon
 */
public Icon getSmallIcon() {
    Icon icon = null;
    try {
        // El icono se encuentra dentro del bundle:
        icon = new ImageIcon(this.getClass().getResource(
            "org/os4os/dskclock/sclock.gif"));
    } catch (Exception ex) {
        icon = new ImageIcon(this.getClass().getClassLoader().getResource(
            "org/os4os/dskclock/sclock.gif"));
    }
    return icon;
}

/**
 * Devuelve el icono de mayor tamaño asociado con la aplicación
 * @return Icon
 */
public Icon getBigIcon() {
    Icon icon = null;
    try {
        // El icono se encuentra dentro del bundle:
        icon = new ImageIcon(this.getClass().getResource(
            "org/os4os/dskclock/clock.gif"));
    } catch (Exception ex) {
        icon = new ImageIcon(this.getClass().getClassLoader().getResource(
            "org/os4os/dskclock/clock.gif"));
    }
    return icon;
}
}
```

Nótese en las líneas subrayadas la clase Launcher. Tal como se aprecia en el método `getInstance`, Launcher debe implementar la interfaz `Command`.

Esta interfaz define la funcionalidad de un comando para el escritorio. Esto es, un comando que puede iniciar la ejecución de una aplicación, etc. `Command` hereda de `Action`, y no es un componente Swing, aunque define el método `getLauncherObject()` que debe devolver un `Object` que bien podría ser un `Component` de Swing. Esto nos permite definir componentes con mucha libertad (nuevos tipos de botones, controles, etc,) pero no vamos a detenernos en esto. Hay una forma más simple de crear un botón lanzador de nuestra aplicación que al fin y al cabo es lo que pretendemos. Esta forma es heredar de `CMenuItem`, una clase abstracta que implementa `Command` y que está definida en el paquete `org.gateway.home.desktop.components` del bundle `desktop`. Heredando de esta clase sólo tendremos que implementar dos métodos: `getCommandName` y `execute`. Siguiendo el ejemplo:

```
package org.os4os.dskclock;

import org.gateway.home.desktop.components.CMenuItem;
import org.gateway.home.desktopbase.components.DesktopInterface;

public class Launcher extends CMenuItem {

    private static final long serialVersionUID = -611567555047776783L;

    private String commandName = "The Clock";

    public Launcher() {
        super("The Clock");
        this.setName("The Clock");
    }

    public String getCommandName() {
        return this.commandName;
    }

    public void execute(DesktopInterface desktop) {
        ClockWindow clock = new ClockWindow();
        clock.setVisible(true);
        desktop.getManager().addFrame(desktop, clock, this);
    }
}
```

Bueno, y aquí ya podemos apreciar la lógica específica de nuestra aplicación. `ClockWindow` es un `JInternalFrame` que se añadirá al escritorio, delegando en el gestor del escritorio (`DefaultDesktopManager`) para ello, mediante el método `addFrame`.

Este método recibe el escritorio al que añadir la ventana, la propia ventana y el lanzador de la aplicación. Y eso es todo. Lo que haga la aplicación ya es asunto del desarrollador. Sólo una apreciación más. ¿Cómo hacemos si queremos proporcionar a la aplicación acceso a los servicios OSGi desplegados en el framework? Muy, muy fácil. A través del gestor del escritorio se proporciona el contexto del bundle:

```
BundleContext context = desktop.getManager().getBundleContext();
```

Una vez hemos obtenido el objeto `BundleContext` podemos pasarlo como parámetro en el constructor de nuestra aplicación:

```
ClockWindow clock = new ClockWindow(context);
```

o en algún otro método


```
clock.setContext(context)
```

Para saber como obtener un servicio OSGi a partir del contexto, consultar la documentación sobre OSGi.

Este enfoque nos ha permitido clasificar aplicaciones, diferenciando si tienen o no interfaz gráfica, e integrarlas con un escritorio y gestor de aplicaciones con interfaz gráfica. Paralelamente a este trabajo, Eclipse llevó a cabo un enfoque muy parecido para su entorno de desarrollo orientado a plataformas OSGi con interfaz de usuario ricas (con escritorios basados en SWT principalmente). El enfoque de Eclipse (Eclipse Rich Client Platform), más completo aunque muy parecido al comentado aquí permite además definir las aplicaciones de forma declarativa a través de un fichero XML [37], lo que supone un paso más allá en la evolución de la clasificación de aplicaciones.

3.5.4. Técnicas de adaptabilidad

En cuanto a la implementación de técnicas de adaptabilidad, es necesario tener en cuenta distintos niveles: adaptabilidad del entorno de ejecución y adaptabilidad de los cursos. El entorno de ejecución debe adaptarse dinámicamente tanto a las necesidades del dispositivo como de los cursos que se instalen en la plataforma. La adaptación a las características del dispositivo se llevará a cabo en el momento de instalación y posterior configuración inicial de los bundles en su fase de arranque. La activación de los bundles deberá averiguar y guardar en una zona de almacenamiento persistente parámetros como la resolución de la pantalla, dispositivos de captura de audio y vídeo (así como los formatos soportados por éstos), idioma preferido del usuario, etc. Este método de comprobación inicial permitirá que un mismo bundle no dependa de una configuración concreta del dispositivo del cliente, adaptándose a éste. La adaptación de los contenidos a las necesidades del usuario requiere ser tenida en cuenta en la fase de desarrollo del curso, de forma que recursos que puedan ofrecerse en varios idiomas (textos, ficheros de audio, etc.) han de ser incluidos en el propio bundle. Es necesario expresar mediante algún mecanismo las diferentes medidas adaptativas posibles, siendo a nuestro juicio el lugar más indicado para ello el manifiesto del paquete educativo (imsmanifest.xml) al igual que se hace en IMS Learning Design, el resultado final de aplicar una regla de adaptación no es sino una organización más de entre todas las posibles. En el siguiente apartado de esta memoria, se describe una aproximación diferente para aplicar reglas de adaptación a los contenidos, si bien creemos que no es la forma más adecuada, puesto introduce la personalización de cada usuario en el curso, lo que es prácticamente imposible en tiempo de desarrollo (se desconocen a qué usuarios irá dirigido). Por ello creemos que, aprovechando la arquitectura orientada a servicios en la que nos apoyamos, la solución puede basarse en la comunicación entre los servicios de administración de usuarios, el de permisos de usuario (hay que tener en cuenta que no todos los usuarios pueden tener permisos para llevar a cabo todo tipo de acciones) y el propio curso. Esto daría lugar a un nuevo servicio de Personalización, que podría admitir la introducción y modificación de reglas de adaptación por parte del usuario, y que teniendo en cuenta el perfil del mismo y las reglas de adaptabilidad contenidas en el curso, aplicase las que coincidan en ambos servicios sin perjuicio de los permisos asociados a dicho usuario. Otra característica deseable es la personalización del entorno de ejecución por parte del usuario, para lo que sería necesario investigar diferentes formas de proporcionar esta funcionalidad. Eclipse ha definido un mecanismo de plug-ins y extensiones en su PDE, que permite la incorporación dinámica de plug-ins en tiempo de ejecución, así como la posibilidad de seleccionar entre varias vistas disponibles. Para entornos web podría delegarse en un procesador de transformaciones XSLT para generar la presentación acorde a las necesidades o preferencias del usuario, y más simple pero no menos eficiente es el uso de hojas de estilo CSS para permitir al usuario la personalización del estilo o tema de su entorno de aprendizaje. Todos estos modos de personalización, no obstante exigen la definición de unos esquemas que identifiquen los diferentes aspectos personalizables y a los que deben ajustarse los diferentes temas gráficos.

En el siguiente capítulo, se describen algunas técnicas aplicadas en el proyecto Passepartout.

3.6. Justificación del proyecto

Estudios realizados en Europa han llevado a la conclusión de que lo aprendido durante los estudios académicos, sobre todo en carreras técnicas, pasa a ser obsoleto en un período de diez años, lo que hace que sea preciso un reentrenamiento para los profesionales, a la vez que se precisa una readaptación para las nuevas tareas que vayan apareciendo, ocasionadas en parte por la rápida evolución de la tecnología.

Por otro lado la UE está tratando de homogeneizar los estudios en todo los países miembros, con una tendencia a la disminución del número de horas lectivas necesarias para la obtención de un título universitario, pero como el Prof. Dr. Guillermo Rodríguez Izquierdo, antiguo Rector de la Universidad Pontificia Comillas dijo: *“es necesario cambiar los métodos educativos, para enseñar las mismas cosas en menos tiempo, sin reducir el nivel, profundidad o calidad del conocimiento, creando un efecto sinérgico entre educación, entrenamiento y vida económica, para optimizar el conocimiento y la especialización”*. [39]

Debido a estas y muchas más razones, la educación académica y la continua deben hacer frente a estos desafíos, encontrando y aplicando *nuevos sistemas* para hacer accesible la educación a estudiantes universitarios y a postgraduados, sin reducir la *calidad de la enseñanza*.

La necesidad de este proyecto puede entenderse fácilmente con el siguiente ejemplo. En muchos ámbitos es necesaria la elaboración de material educativo personalizado. Por ejemplo, no sólo varían lo contenidos, sino que la forma de presentar los contenidos también difiere según el tipo de audiencia al que va destinado. Por ejemplo, la forma de presentar los contenidos para un curso de primaria, uno de secundaria, o uno de enseñanza superior es totalmente diferente. Igualmente la personalización de esta presentación difiere en cursos especializados para nuevos conductores, para bomberos, policías, militares, etc. Típicamente, cada organización o centro encargado de estas áreas, diseñan sus propios materiales o pagan a un tercero por el diseño e implementación de estos materiales. Esta práctica da como resultado la existencia de numerosas organizaciones invirtiendo por desarrollar material educativo una y otra vez, para audiencias que difieren únicamente en pequeños detalles.

Esto hace que aparezcan dependencias: con respecto al formato de distribución seleccionado (CD-ROM, transparencias, DVD, cintas de vídeo, cintas de cassette, papel, etc.), del sistema operativo, de la herramienta de gestión de aprendizaje utilizada, de las herramientas de autor utilizadas para la elaboración de los contenidos, etc. Un cambio en alguno de estos componentes en un curso podría tener un fuerte impacto en el resto de componentes del curso y requerir la creación de una nueva versión o modificar el sistema completo.

Supongamos que el cincuenta por ciento del total del trabajo realizado por estas organizaciones fuera genérico para cualquier audiencia y pudiera ser utilizado por cualquiera de las organizaciones encargadas del diseño. Asumamos que el coste de esa porción genérica del desarrollo es de 10.000 € y que cada organización gasta otros 10.000 € en personalizarlo para su audiencia específica (conductores, bomberos, militares, alumnos universitarios, etc.). Si hay diez escuelas de conducción, diez academias de bomberos, cuatro ramas militares y diez academias universitarias que necesitan aplicaciones específicas, hacen un total de 34 organizaciones que gastarían un total de 680.000 € (20.000 € cada una).

Si hubiera una vía para que esas organizaciones pudieran compartir esa parte genérica, podrían ahorrar 330.000 €:

10.000 € de la parte genérica + 34 organizaciones x 10.000 € de personalización = 350.000 €.

De esta forma se pagarían los gastos de la parte genérica entre todas las organizaciones, pero sólo una vez, comprometiéndose todas las organizaciones a compartir esta porción.

De ahí que sea importante definir sobre una arquitectura abierta y libre de ser usada por cualquier organización, un framework para eLearning, abierto y de código libre.

Pero no todo es tan fácil, se podría pensar. Cada organización tiene unos requisitos distintos: distintos sistema operativo (Windows, Solaris, Linux, McIntosh, etc.), distintas herramientas de autor, distintos medios de difusión (internet, Tv digital, PDAs, telefonía móvil, automóviles), etc. Son estos detalles los que justifican el uso de OSGi, una iniciativa que define un framework abierto y que, al haber sido especificado enteramente en Java, lo hace independiente de la plataforma. Además está pensado para poderse empotrar en dispositivos como teléfonos móviles, automóviles, pasarelas residenciales, receptores de TV digital, etc.

De esta forma, apoyándonos sobre el framework de OSGi, nos beneficiamos de las ventajas de esta iniciativa, ampliamente apoyada por importantes socios industriales, y conseguimos una parte genérica de nuestro sistema de eLearning abierta y capaz de integrarse y comunicarse con prácticamente cualquier dispositivo que cumpla la especificación OSGi.

Para esta justificación sólo nos hemos apoyado en cuestiones económicas y técnicas, pero existen muchas más razones que apoyan esta investigación y que atienden a factores pedagógicos, éticos, de interfaces gráficas, institucionales, administrativos, etc.

A continuación se comentan diversos aspectos relacionados con la investigación, frutos de la participación en algunos proyectos de investigación europeos.

4. Proyectos

A continuación se describen los proyectos relacionados con esta investigación en los que el autor ha participado y los que se ha experimentado con algunas posibles soluciones a los problemas planteados.

4.1. EDUKA



Eduka (2002) fue presentado como Proyecto Fin Carrera del autor de esta memoria, recibiendo el Premio "Mejor Proyecto Fin de Carrera Patrocinado por Fidetia" en el año 2003. Eduka era una aplicación que podía integrarse tanto en applets para ejecutarse vía web, como ejecutarse en local desde la línea de comandos como una aplicación de ventanas convencional, pero que en realidad se trataba de una aplicación web distribuida que ofrecía una serie de servicios a través de la red. Dichos servicios fueron especialmente diseñados para ser utilizados como medios de transmisión y recepción de señales de audio y vídeo, mensajes de texto, etc., que facilitasen la comunicación entre la parte transmisora (profesor en el aula) y los receptores (alumnos en sus casas, otro centro, etc.)

A diferencia de la típica arquitectura de tres capas de estas aplicaciones, la arquitectura de Eduka incluye una cuarta capa más, en la que se sitúan los servicios remotos ofrecidos por la aplicación y con los que se comunica el servidor web. Sin embargo estos servicios no se apoyaban sobre ningún framework que facilitase la gestión de su ciclo de vida (como podría ser OSGi, Java Services Framework o Spring) sino que únicamente registrabas esos "servicios" en un registro RMI conocido. Esta nueva abstracción posibilitaba que la lógica de la aplicación pudiera distribuirse en una red local y no quedar centralizada en un único servidor. Además suponía un punto de partida para la multiplicidad de servidores y pasar de un sistema monoservidor, a tener un sistema dual o con tantos servidores especializados como fueran necesarios (servidor de vídeo, servidor documental, servidor de expedientes, etc.). Tan sólo era necesario modificar una serie de ficheros de configuración en los que se indicaban qué servidor se ocupaba de cada servicio. Sin embargo, los únicos estándares que se tuvieron en cuenta para el proyecto estuvieron relacionados con los protocolos de transmisión en tiempo real RTP y RTSP, y la metodología de aprendizaje sugerida se componía de una mezcla de multi-conferencias (videoconferencias mediante las que la clase impartida por el profesor se retransmitía a una red de multidifusión a la que los alumnos podían conectarse) con soporte para el intercambio de opiniones (chat) y de ficheros, las tutorías electrónicas y un almacén de documentos clasificados según materias y niveles educativos. Sin embargo, no se tuvo en cuenta en el proyecto aspectos como la descripción de los cursos mediante meta-datos y las especificaciones (2001 y 2002) que en aquel tiempo comenzaban a publicar sus primeros borradores como SCORM 1.1 o IMS CP.

Además del Premio al Mejor Proyecto Fin de Carrera, el trabajo de investigación relacionado con el proyecto Eduka, "Aplicación de Sistemas Hipermedia al Desarrollo de Clases Virtuales y Semipresenciales" recibió el "Premio a la Innovación Docente de la Universidad de Sevilla. Curso 2001/2002". Este éxito con Eduka supuso para el autor el punto de partida y la motivación por la investigación acerca de los sistemas de eLearning, que lo llevó a matricularse en los cursos de Doctorado en Lenguajes y Sistemas Informáticos.

4.2.OSMOSE



OSMOSE⁵⁸ (Open Source Middleware for Open Services in Europe) fue un proyecto de dos años de duración (2003 – 2005) enmarcado dentro del programa Eureka-ITEA (código ITEA 02003) financiado por la Unión Europea a través de las Administraciones Públicas de cada país. En España, fue financiado por el Ministerio de Ciencia y Tecnología a través del Programa de Fomento de la Investigación Técnica (PROFIT) y por el Ministerio de Industria, Turismo y Comercio mediante el Programa 542N-Investigación y Desarrollo de la Sociedad de la Información. Ocho países de la Unión Europea estuvieron representados en el proyecto, en el que participaron seis grandes empresas, Bull, France Télécom, Philips, Telefónica, TELVENT y Thales; pequeñas y medianas empresas como Bantry Technologies, iTEL, Kelua, Lynx, VICORE y Whitestein Technologies; y centros de investigación y universitarios como Charles University of Prague, EPFL⁵⁹, INRIA⁶⁰, INT⁶¹, LIFL⁶², LSR⁶³ y DIT⁶⁴.

El propósito de OSMOSE fue definir un conjunto de elementos hardware y software de código libre, que facilitase la comunicación entre dispositivos en el entorno residencial, así como que garantizase, en la medida de lo posible, la ejecución de aplicaciones multimedia y su integración con la TV digital. El proyecto se dividió en tres paquetes de trabajo principalmente. En el primero se encaminó la investigación a la definición un modelo de componentes para la construcción de un middleware abierto basado en este modelo de componentes. En el segundo paquete de trabajo, se trató de demostrar la validez de este modelo en tres plataformas tecnológicas distintas: J2EE, OMG CCM y OSGi. Y finalmente en el tercero, se construyeron demostradores o pilotos, que para demostrar de forma práctica la aplicabilidad de los resultados de los paquetes anteriores. En OSMOSE se concibe la pasarela residencial como nexo de unión entre Internet y la red del hogar, tal y como se aprecia en la figura siguiente.

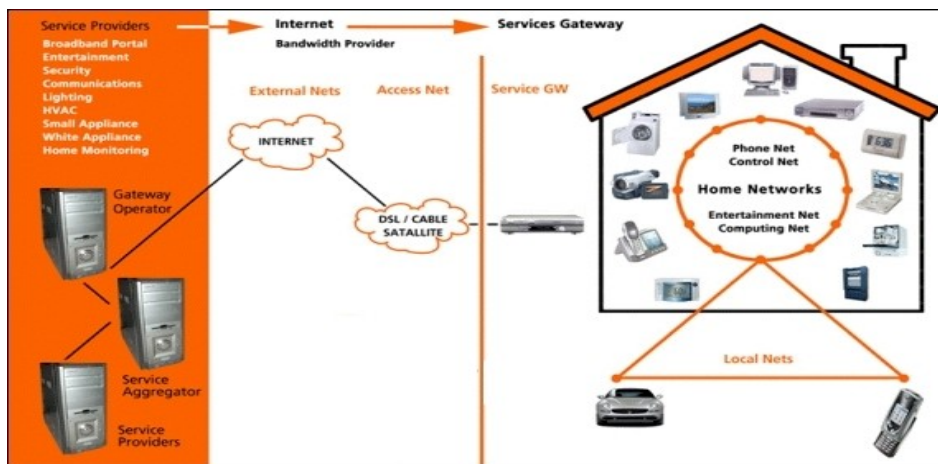


Figura 29. Escenario de OSMOSE: el hogar inteligente

⁵⁸ <http://www.itea-osmose.org>

⁵⁹ Ecole Polytechnique Fédérale de Lausanne, <http://www.epfl.ch/>

⁶⁰ Institut National de Recherche en Informatique et en Automatique, <http://www.inria.fr/>

⁶¹ Institut National des Télécommunications, <http://www.int-evry.fr/>

⁶² Laboratoire d'Informatique Fondamentale de Lille, <http://www.lifl.fr/>

⁶³ Laboratoire Logiciels, Systèmes et Réseaux, Institut d'Informatique et Mathématiques Appliquées de Grenoble. <http://www-lsr.imag.fr/>

⁶⁴ Departamento de Ingeniería Telemática. Universidad Politécnica de Madrid. <http://www.dit.upm.es>

La figura anterior representa un posible escenario de aplicación del proyecto OSMOSE, el hogar inteligente, en el que las aplicaciones domóticas juegan un papel importante a la hora de informar del contexto. En este escenario, la pasarela residencial constituye el nexo de unión entre el mundo exterior y el interior, hace a la vez de enrutador y de contenedor de servicios para otros dispositivos interconectados a la pequeña red local del hogar. En este escenario pueden convivir diferentes versiones de eLearning: tLearning, mLearning, CBT, etc., pero la investigación acerca relacionada con los estándares eLearning no es el objetivo prioritario de este proyecto, sino la definición de un middleware escalable, que facilite, entre otras, este tipo de aplicaciones informáticas.

Se eligió una especificación abierta de arquitectura orientada a servicios que ha experimentado un notable auge y aceptación tanto en la industria como en la comunidad de software libre: OSGi.

La arquitectura de referencia de OSGi define un framework para controlar el ciclo de vida de los servicios en él instalados. En un principio ideada para sistemas empotrados, las ventajas que ofrece para el control en tiempo de ejecución de los servicios ha hecho que sea adoptado como framework base para una gran variedad de aplicaciones. Varios ejemplos de la diversidad de aplicaciones y la importancia que la iniciativa de OSGi ha tenido en la industria son el hecho de que el entorno de desarrollo Eclipse haya sido re-diseñado (a partir de su versión 3.2) tomando como base una implementación propia del framework de OSGi: Equinox⁶⁵, que BMW utilice un framework OSGi para la gestión de datos del vehículo [32], y que Apache esté en desarrollo de su propia implementación del framework OSGi, Apache Felix⁶⁶, tomando como punto de partida la implementación desarrollada por Richard S. Hall: OSCAR⁶⁷. Otras implementaciones open source del framework de OSGi como Knopflerfish⁶⁸ también ha tenido una gran repercusión en este campo.

En OSMOSE se optó por OSCAR como framework OSGi, pues en el proyecto contamos con la colaboración de su autor, Richard S. Hall, a través del LSR⁶⁹ de la Universidad de Grenoble, miembro del consorcio del proyecto, con el que colaboramos en el desarrollo de OSCAR y algunos servicios básicos como el servicio de Configuración, el de Log, etc.

Los principales temas abordados por el autor de esta memoria en este proyecto se centraron en:

- Despliegue de servicios y gestión de dependencias
- Publicación de servicios
- Integración de TV y aplicaciones interactivas

Despliegue de servicios y gestión de dependencias

La unidad de despliegue de servicios en OSGi es el "bundle". Un "bundle", es un archivo .jar (Java ARchive) cuyo contenido puede estar compuesto por clases Java, librerías nativas y otro tipo de ficheros (imágenes, archivos de texto...) y que usa el fichero Manifiesto definido en la especificación de los archivos Java [33], añadiéndole nuevas propiedades para describir los paquetes o clases que deberán exportarse para que otros bundles puedan usarlos, el autor o autores y la versión del bundle, los paquetes o clases proporcionados por otros bundles que deben ser importados, etc. La funcionalidad para una aplicación concreta puede requerir de uno o más servicios OSGi repartidos en uno o más bundles, estableciendo de esta forma dependencias entre

⁶⁵ <http://www.eclipse.org/equinox/>

⁶⁶ <http://cwiki.apache.org/FELIX/index.html>

⁶⁷ <http://oscar.objectweb.org/>

⁶⁸ <http://www.knopflerfish.org/>

⁶⁹ Laboratoire Logiciels, Systèmes et Réseaux, recientemente integrado en el Laboratoire d'Informatique de Grenoble (LIG), <http://lig.imag.fr/>

los bundles implicados. Estas dependencias se deben a las necesidades de importación de clases de otros bundles, puesto que OSGi obliga al uso de un cargador de clases para cada bundle, de forma que para cada bundle sólo serán visibles las clases incluidas en él, siendo el framework el que se encargará de pre cargar y tener disponibles las referencias a las clases que dicho bundle declara necesarias para importar.

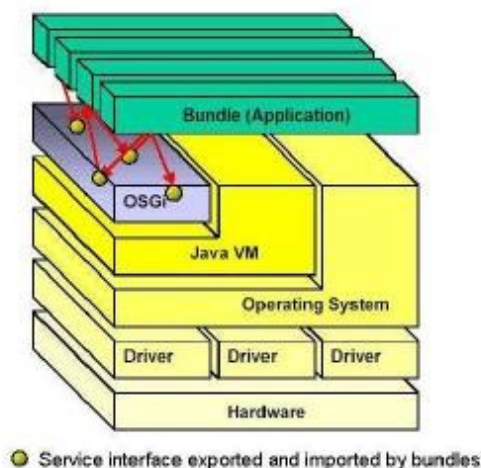


Figura 30. Arquitectura básica de una pasarela de servicios OSGi

```
Manifest-Version: 1.0
Ant-Version: Apache Ant 1.6.5
Created-By: 1.4.2_03-b02 (Sun Microsystems Inc.)
Bundle-ManifestVersion: 2
Bundle-Version: 2.1.0
Bundle-ContactAddress: http://www.telvent.com
Bundle-Name: Desktop Base
Bundle-Description: Base classes for Telvent's Osmose Desktop
Bundle-DocURL: http://www.telvent.com
Bundle-Vendor: Telvent
Bundle-SymbolicName: Desktop Base
Bundle-Category: service
Export-Package: org.gateway.home.desktopbase.service,
                org.gateway.home.desktopbase.utils,
                org.gateway.home.desktopbase.components,
                org.gateway.home.desktopbase.applications.shellgui
Import-Package: org.osgi.framework
Bundle-Activator: org.gateway.home.desktopbase.service.Activator
```

Listado 3. Ejemplo de Manifiesto de un bundle

En el Manifiesto del listado anterior se especifica una dependencia con el paquete `org.osgi.framework`. Esto quiere decir que en el bundle que describe este Manifiesto, se usan clases del framework de OSGi. Esto es lo normal para bundles que no sean simplemente librerías, pues

han de contener una instancia de `org.osgi.framework.BundleActivator`, tal y como indica la propiedad `Bundle-Activator` del listado anterior. Esta instancia controla el arranque y parada del bundle y tal y como se puede apreciar en el listado siguiente se encuentra en el paquete `org.osgi.framework` y además incluye una dependencia con la clase `BundleContext` del mismo paquete.

```
package org.osgi.framework;

public interface BundleActivator {

    public void start(BundleContext context) throws Exception;

    public void stop(BundleContext context) throws Exception;

}
```

Listado 4. Interfaz BundleActivator

De esta misma forma otro bundle podría tener dependencias con el anterior, usaría clases exportadas por éste, para lo que necesitaría importarlas, tal y como se muestra en el siguiente listado.

```
Bundle-ManifestVersion: 2
Created-By: 1.4.2_03-b02 (Sun Microsystems Inc.)
Bundle-Activator: org.gateway.home.desktop.DesktopBundleActivator
Bundle-ContactAddress: http://www.telvent.com
Bundle-Name: Desktop
Bundle-Description: Osmose Telvent Desktop
Bundle-DocURL: http://www.telvent.com
Bundle-Vendor: Telvent
Bundle-SymbolicName: Desktop
Bundle-Category: service
Import-Package: org.osgi.framework, org.gateway.home.desktopbase.utils,
               org.gateway.home.desktopbase.service,
               org.gateway.home.desktopbase.components,
               org.gateway.home.desktopbase.applications.shellgui,
               org.ungoverned.osgi.service.shell
               org.ungoverned.osgi.service.bundlerepository,
               org.osgi.service.log, info.clearthought.layout
Export-Package: org.gateway.home.desktop,
               org.gateway.home.desktop.components, resources.images
Bundle-Version: 2.1.0
```

Listado 5. Manifiesto del bundle desktop.jar con que depende del anterior

En OSGi las dependencias son a nivel de clases, pero en OSMOSE planteamos un nuevo nivel de abstracción: no desplegar bundles (cuyas dependencias son clases) sino considerar el despliegue de aplicaciones funcionalmente completas que pueden estar compuestas por uno o más bundles, y

cuyas dependencias pueden incluir, no sólo al conjunto de bundles que forman la aplicación, sino a las dependencias de cada uno de ellos (que pueden ser también con paquetes y librerías nativas), y así sucesivamente hasta resolver por completo todas las dependencias. Un problema parecido al que resuelve el gestor de paquetes de las distribuciones Linux basadas en Debian⁷⁰, apt-get, a la hora de desplegar nuevos paquetes.

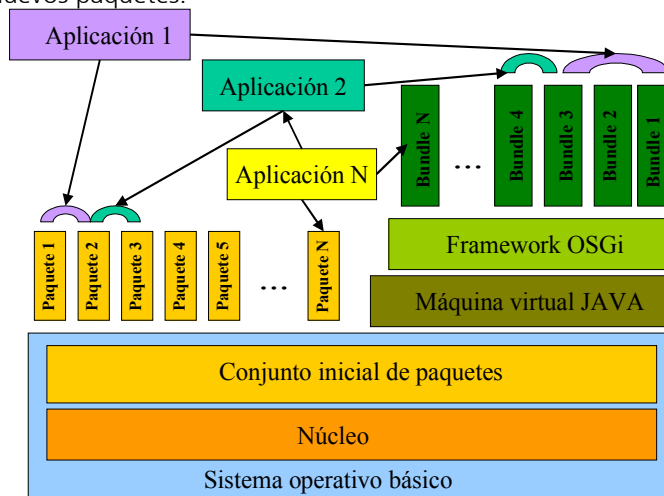


Figura 31. Esquema de modelo de dependencias entre aplicaciones

Este problema nos llevó a diseñar un mecanismo avanzado de despliegue [35] encargado de la resolución de los posibles conflictos derivados de las dependencias y del despliegue de todos los bundles necesarios, desarrollándose para ello varias herramientas que ayudaron a dividir el problema, tal y como se muestra en el siguiente diagrama:

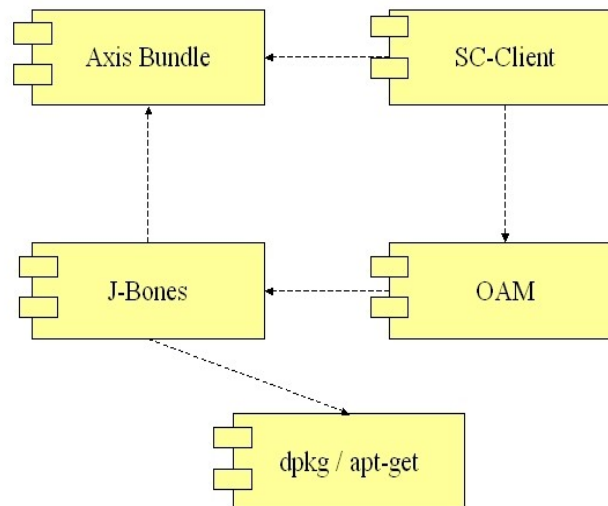


Figura 32. Componentes del mecanismo de despliegue

Los componentes que intervienen son:

⁷⁰ <http://www.debian.org/>

Axis Bundle⁷¹: Bundle desarrollado por Telvent en el proyecto OSMOSE y que envuelve la funcionalidad de Apache Axis.

SC-Client⁷²: Services Catalogue-Client. El cliente del Catálogo de Servicios, es una aplicación residente en la pasarela que muestra al cliente las aplicaciones disponibles para desplegar en su pasarela, teniendo en cuenta las características de la misma. Se comunica con servidor del Catálogo de Servicios SC-Server haciendo uso de servicios web, para lo cual usa la funcionalidad proporcionada por Axis Bundle.

J-Bones⁷³: Herramienta para resolver dependencias a nivel de bundles y paquetes nativos. Desarrollada por el Departamento de Ingeniería Telemática de la Universidad Politécnica de Madrid en colaboración con Telvent. Usa la herramienta de despliegue de paquetes nativos de Debian, dpkg, así como Axis Bundle para comunicarse con servicios remotos a través de servicios web.

OAM⁷⁴: Osgi Application Manager. Herramienta desarrollada por Telefónica I+D dentro del proyecto OSMOSE y que se encarga de resolver las dependencias a nivel de aplicaciones, delegando en J-Bones para la resolución a nivel de bundles. Recibe el descriptor de la aplicación a instalar de parte del Cliente Catálogo de Servicios.

Estos componentes están presentes en la pasarela residencial, dónde se lleva a cabo la resolución de dependencias, previa obtención del descriptor o descriptors de la aplicación del Centro de Control. Aunque pudiera parecer más lógico, que fuera en el lado del servidor (del Centro de Control de pasarelas) dónde se debiera llevar a cabo la resolución de dependencias, ésta se realiza en el cliente por dos motivos principalmente:

-Para posibilitar la instalación de componentes sin contar con conexión a la red. Por ejemplo instalar aplicaciones desde un CD-ROM o memoria flash USB.

-Algunas aplicaciones pueden contar con dependencias de aplicaciones que ya están instaladas en la pasarela, y de resolución inmediata, evitando así la comunicación entre servidor y pasarela, para informar de estos casos.

Este mecanismo utilizaba un fichero escrito en XML que servía de descriptor de la aplicación, y cuya ubicación era conocida, en el centro de control (Gateway Operator en la Figura 16) de las pasarelas residenciales, por lo que los proveedores de servicios (encapsulados en aplicaciones desplegables en las pasarelas) debían de proporcionar al Centro de Control estos descriptors.

Tal y como se muestra en [35] el descriptor de la aplicación contiene tanto información descriptiva de la aplicación, comprendida entre las etiquetas `<information>` y `</information>`, útil al cliente para decidir la instalación o no de la misma como dependencias a nivel de aplicación contenida entre las etiquetas `<resources>` y `</resources>` indicando la ubicación de los descriptors de las aplicaciones de las cuales depende, así como los bundles que componen la aplicación descrita. La solución a nivel OSGi resultó bastante satisfactoria, sin embargo los paquetes nativos o quedaban totalmente integrados en el modelo, debiéndose buscar una solución análoga. El objetivo es que OAM asocie a las aplicaciones las dependencias de estas hacia los paquetes nativos de la misma forma que asocia los bundles, es decir no sólo para instalar o desinstalar, sino también para arrancar y parar los procesos asociados a estos paquetes junto con el resto de la aplicación.

⁷¹ <http://forge.os4os.org/projects/ws-support/>

⁷² <http://rtdwork.org/projects/scclient/>

⁷³ <http://jbones.forge.os4os.org/>

⁷⁴ <http://forge.os4os.org/projects/tid-oam/>

Publicación de servicios

El problema de publicación de servicios, se afrontó con la necesidad de contar con un mecanismo que permitiese la integración de las pasarelas con centros de control de diferentes tecnologías, dado el carácter del proyecto en el que se enmarca (recordamos el objetivo de demostrar la interoperatividad entre tres plataformas diferentes). Para ello se decidió el uso de servicios web para la comunicación de la pasarela tanto con los dispositivos del hogar como con el centro de control, definiéndose la funcionalidad necesaria en ambos lados, tal y como se expone en [36]:

En el Centro de Control, es necesario contar un Servicios Web que permita consultar las aplicaciones disponibles, admitiendo opciones de filtrado (por tipo de pasarelas, por perfil de usuario, etc.) Además es necesario también hacer accesible mediante Servicios Web la funcionalidad que necesitan los proveedores de servicios para añadir nuevas aplicaciones, según el modelo de negocio definido en el proyecto.

Por otra parte en la pasarela, una solución para las aplicaciones destinadas a ser usadas o accedidas desde otros dispositivos, consiste en publicar los servicios OSGi como servicios web mediante una implementación de SOAP en la pasarela, para lo cual se usó el componente AxisBundle.

Para facilitar la publicación de los servicios OSGi como servicios web, AxisBundle registra un servicio OSGi en el framework definido por la interfaz `AxisBundleFunctionality`, cuyos métodos son:

```
public void publishService(String serviceName, String serviceClassName,
    String provider, String allowedMethod,
    BeanMapping[] mapping,
    AxisCustomizingMapping[] serializers,
    GenericParam[] params)
    throws PublishServiceException;

public void publishService(String serviceName, String allowedMethod,
    BeanMapping[] mapping,
    AxisCustomizingMapping[] serializers,
    String osgiService)
    throws PublishServiceException;

public void removeService(String serviceName)
    throws PublishServiceException;
```

Listado 6. Métodos de la interfaz `AxisBundleFunctionality`

El primer método permite publicar servicios web que no son necesariamente servicios OSGi, esto es, que no se registran en el framework de OSGi como tales.

El segundo método es el que se debe usar para registrar como servicio web un servicio OSGi, indicando en el último parámetro el identificador con el que se ha registrado en el framework. El modo de registro con este segundo método es diferente, pues a diferencia del primero no se crea una nueva instancia cada vez, sino que la instancia es única y está registrada en el framework de OSGi, por lo que será el framework quién provee dicha instancia (por lo que el objeto `provider` es innecesario igualmente, puesto que es conocido).

El último método, elimina el registro de un servicio web, independientemente de cómo fuera publicado. Los tres métodos lanzan una instancia de la excepción `PublishServiceException` en caso de producirse algún error.

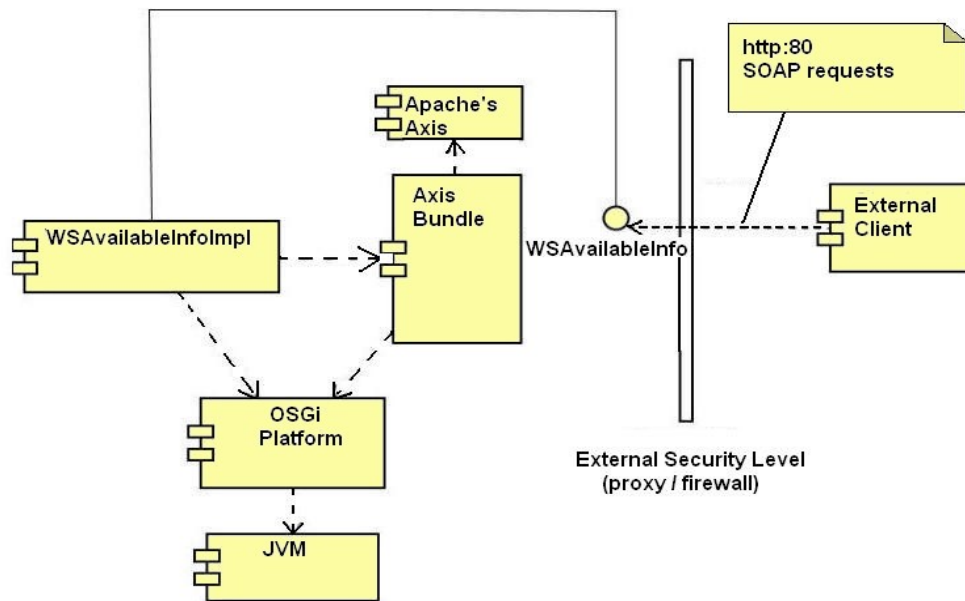


Figura 33. Comunicación de AxisBundle con el resto de componentes de la pasarela

El diagrama anterior muestra la relación entre AxisBundle y el resto de componentes de la pasarela, incluido el propio framework de OSGi, y como un cliente externo sólo conoce los métodos definidos por la interfaz web `WSAvailableInfo`, que son:

- `getPublishServicesList()`; que devuelve la lista de los servicios web publicados en la pasarela.
- `getServiceWsdldescription(Object serviceName) throws ServiceWsdldescriptionException`; que obtiene el descriptor WSDL del servicio web cuyo nombre se pasa.

Además del registro y publicación de servicios OSGi servicios Web, y la interfaz para consultar a la pasarela acerca de los Servicios Web que en ella hay publicados, también se implementó en AxisBundle un mecanismo para invocar remotamente, a través de SOAP a los Servicios Web, tanto locales como remotos, desde cualquier servicio OSGi. Para ello se definió e implementó la interfaz `RemoteServiceCaller`, implementada, al igual que las anteriores en AxisBundle y registrada como servicio OSGi en el framework para que desde cualquier bundle instalado en el mismo pueda accederse a un servicio web externo. Esta interfaz sólo consta de un método:

```
public Object invokeRemoteService (URL endPoint, String serviceName,
                                   String method, ParameterBean[] callParams,
                                   ParameterBean returnParam)
    throws ServiceNotReachableException;
```

Integración de TV y aplicaciones

Tras intentar la integración por software con pésimos resultados, optamos por la incorporación de un procesador dedicado, diseñado por Philips: el Trimedia.

Para la integración se definió un modelo de datos que diferenciara las aplicaciones sin interfaz gráfica de usuario de las aplicaciones con ella, diferenciando para las últimas un nuevo ciclo de vida gestionado por un servicio similar a los tradicionales gestores de ventanas de Windows, Gnome, o Kde...

El escritorio, registraba un servicio denominado `GUIService`, que implementa las interfaces `org.osgi.framework.ServiceListener` y aplicando la filosofía del patrón Observador, se encarga de detectar en los eventos de cambio en el registro de servicios del framework de OSGi (cuando se registra o se elimina del registro un servicio, así como cuándo se actualiza o modifica). Cuando estos eventos correspondían a servicios que implementaban la interfaz `GUIApplication`, es decir, servicios que contaban con interfaz gráfica de usuario, la información proporcionada en los métodos de `GUIApplication` era procesada para añadir los iconos, lanzadores y demás información relativa a dicho servicio al escritorio.

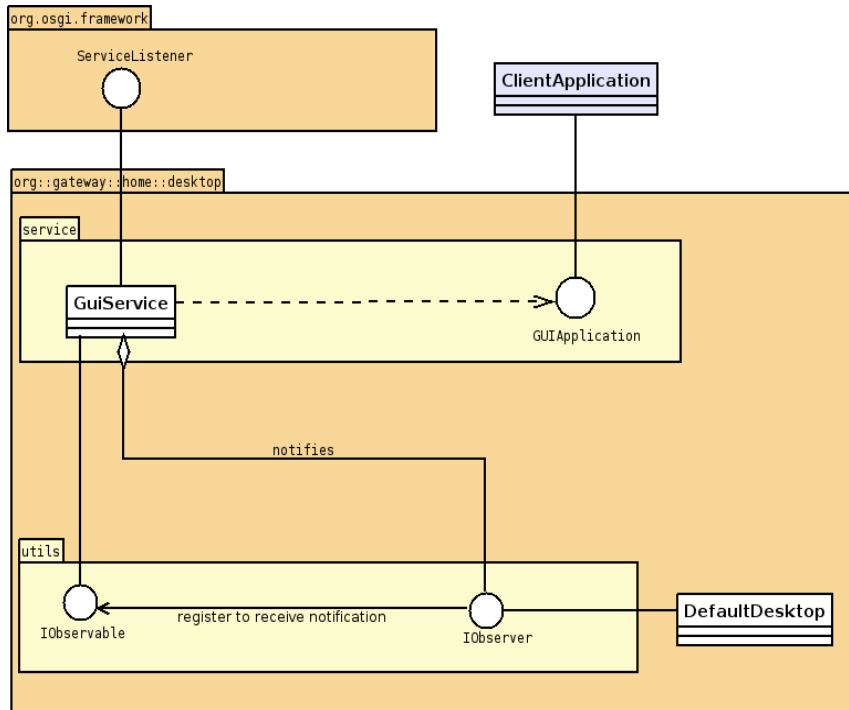


Figura 34. Aplicación del Patrón Observador en el diseño del Escritorio

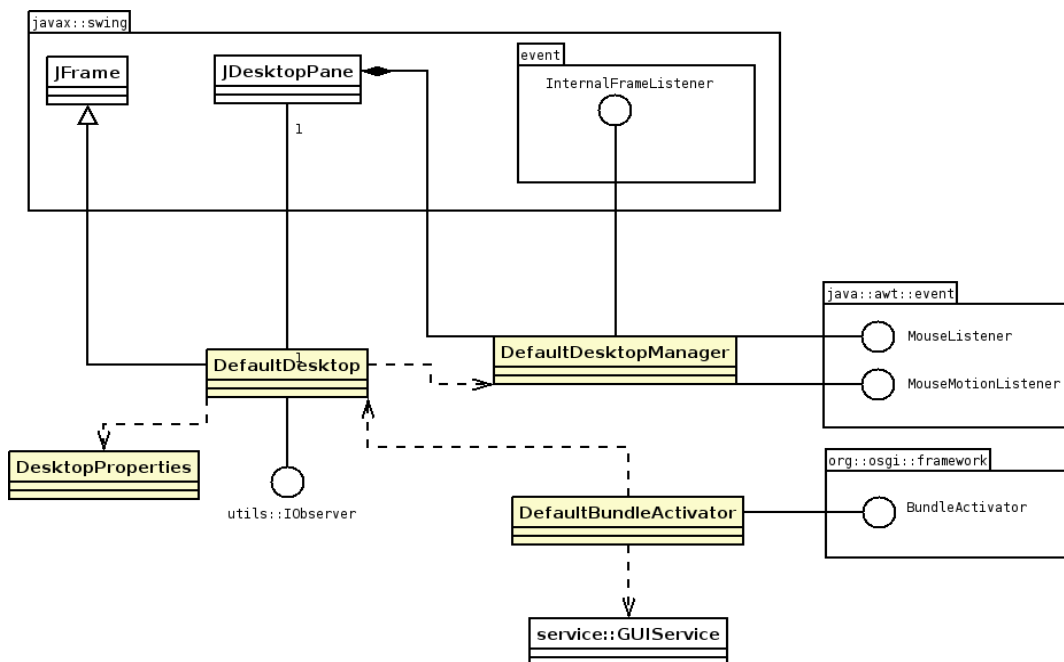


Figura 35. Servicio GUIService y su relación con el Escritorio

Además el Escritorio diseñado para la integración de TV y aplicaciones multimedia en la pasarela residencial, exportaba el paquete `components`, en el que se definían unos componentes abstractos que otros desarrolladores podían usar para añadir a su aplicación lanzadores en la barra de menús, iconos de escritorio, etc. Para ello se definió la interfaz `Command` que permitía implementar órdenes, y que ciertos componentes como los lanzadores `CMItem` implementaban. Para ello se implementó el Patrón de Comportamiento Comando:

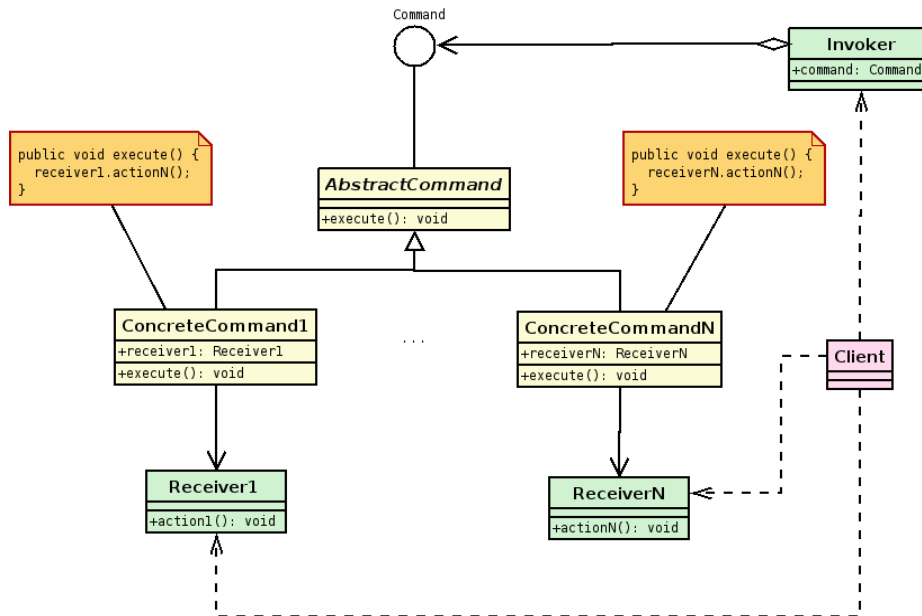


Figura 36. Patrón Comando.

De esta forma, los desarrolladores de aplicaciones con interfaz gráfica encontrarán muy fácil la creación de botones lanzadores de su aplicación, que se añadiría automáticamente al menú del escritorio, y para los que sólo era necesario heredar de la clase abstracta `CMItem` e implementar el método `execute`, como en el ejemplo siguiente:

```

import org.gateway.home.desktop.*;

public class MyCMItem extends CMenuItem{
public CMItemExit() {
    super();
    setEnabled(true);
}
public void execute(DefaultDesktopManager odm) {
    // Añadir acciones de arranque de la aplicación
    // Típicamente: Crear la ventana con el GUI
    // Añadirla al escritorio==>Usar métodos de odm
}
}
    
```

Una vez finalizado el proyecto, conseguimos una pasarela residencial basada en una arquitectura abierta orientada a servicios, en la que instalar aplicaciones domóticas accesibles a través de Servicios Web, con la funcionalidad necesaria para acceder a servicios web remotos cuya ubicación fuera conocida y con un escritorio que facilitaba el desarrollo de aplicaciones con interfaz gráfica de usuario basada en Swing. Algunas de las aplicaciones domóticas que se desarrollaron en esta plataforma para el demostrador final del proyecto, permitieron controlar luces, persianas, así como una simulación de detector de intrusos a través de un servicio de detección de movimiento basado en un algoritmo de detección de flancos en las imágenes capturadas desde una cámara web conectada a la pasarela. Otros miembros del consorcio del proyecto OSMOSE desarrollaron bundles para gestionar el perfil de usuario, así como guardarlo en tarjetas JavaCards, permitiendo así el arranque de un perfil u otro según la tarjeta introducida, personalizando de esta forma la funcionalidad de la pasarela residencial según el perfil de usuario. El resultado final de la pasarela de OSMOSE puede verse en las siguientes imágenes.

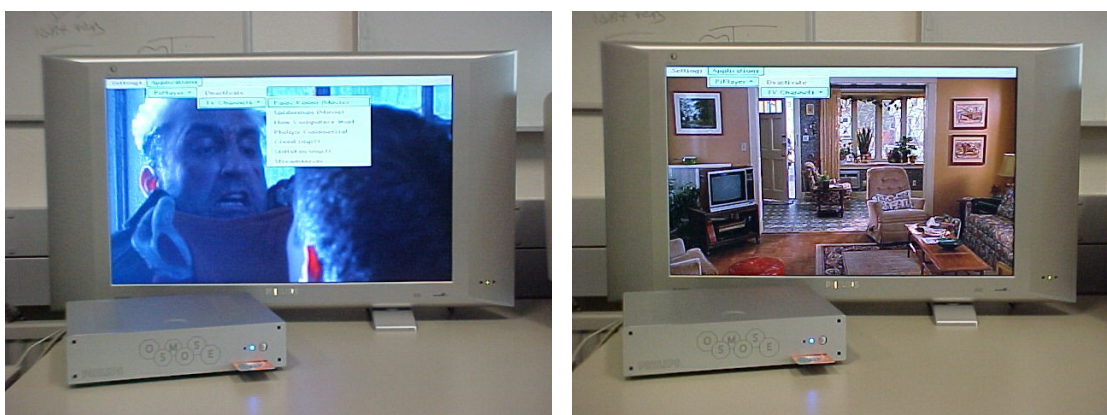


Figura 37. Apariencia externa final de la pasarela

4.3.Passepartout



Passepartout ha sido un proyecto de dos años de duración (2005 – 2007) perteneciente al programa Eureka-ITEA (código ITEA 04017) de la Unión Europea y financiado por Ministerio de Ciencia y Tecnología a través del programa PROFIT (proyecto FIT-330300-2005-28) y cuya revisión final tuvo lugar en el Centro de Conferencias de Philips, en Eindhoven, el pasado 21 de Marzo de 2007. El proyecto Passepartout, propuesto por Philips y Thomson, contó con la participación de TELVENT, Universidad Politécnica de Madrid, Universidad de Vigo, V2⁷⁵, CWI⁷⁶, Prewise, Stoneroos, CRP Henri Tudor⁷⁷, LORIA⁷⁸ y ETRI⁷⁹.

El subconsorcio español en Passepartout, formado por Universidad Politécnica de Madrid, TELVENT y Universidad de Vigo, se centró principalmente en el desarrollo de una plataforma multimedia abierta, apta para t-learning, estudiando las necesidades y posibles soluciones para el aprendizaje a través de canales educativos personalizados, capaces de recomendar contenidos audiovisuales en base a las preferencias (cambiantes) del usuario.

⁷⁵ Institute for the Unstable Media, <http://www.v2.nl>

⁷⁶ Centrum voor Wiskunde en Informatica, <http://www.cwi.nl>

⁷⁷ Centre de Recherche Public Henri Tudor, <http://www.tudor.lu>

⁷⁸ Laboratoire Lorrain de Recherche en Informatique et ses Applications, <http://www.loria.fr>

⁷⁹ Electronics and Telecommunications Research Institute, <http://www.etri.re.kr>

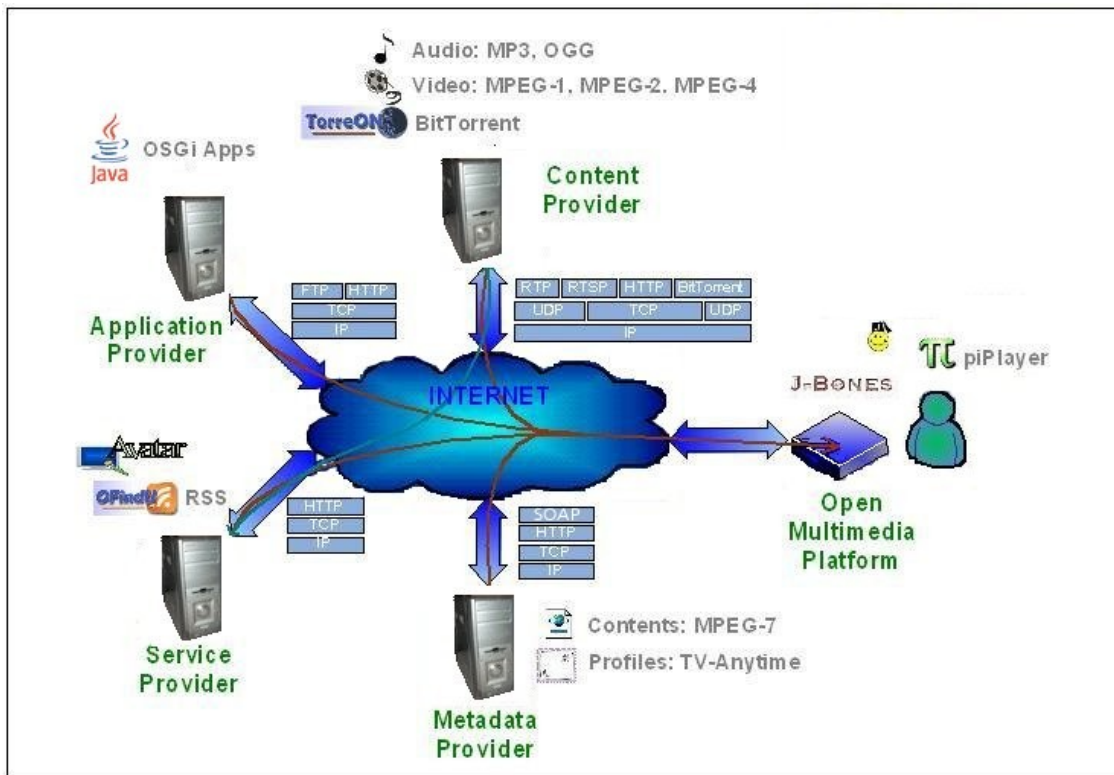


Figura 38. Arquitectura y tecnologías usadas en Passepartout

A la hora de desarrollar cursos aptos para ser difundidos a través de televisión digital, es necesario tener en cuenta ciertas características, tanto técnicas como sociológicas.

En cuanto a características técnicas:

- La baja capacidad de proceso de los set-top-boxes comparado con los PCs convencionales
- Los modos de interacción con un mando a distancia son más pobres.
- La menor resolución de un televisor comparada con la de un monitor.
- La distancia del usuario es mayor frente a un televisor que frente a un monitor de PC.
- Finalmente, el modo en que llega la información: broadcast frente petición bajo demanda.

Y en el plano sociológico:

- Muchos usuarios de televisión tienen un nivel de preparación y predisposición para aprender el uso de nuevas tecnología más bajo que los usuarios de Internet.
- La actitud de los usuarios de TV es pasiva frente a la actitud activa de los usuarios de ordenador.
- Los espectadores de TV conciben la televisión como un medio de entretenimiento.

A diferencia de los cursos que se distribuyen en Internet para PC, en los que predominan texto e imágenes, en los cursos para televisión digital interactiva deben predominar los contenidos audiovisuales. Una vía de atraer a los usuarios hacia los cursos es mediante actividades de entretenimiento relacionadas con los temas que ellos encuentran interesantes, que pueden ser deducidas, por ejemplo, del tipo de programas de TV que ellos ven. Para ello es necesario conocer en cada instante qué usuario está usando la plataforma multimedia, qué contenido está viendo, y qué nivel de aceptación otorga a dicho contenido.

En este proyecto el LMS se implementó en Java, y proporcionando un punto de entrada como servicio OSGi en la plataforma multimedia, no contemplándose la comunicación entre plataforma y LMS remoto, y no teniendo en cuenta, por tanto, un seguimiento del aprendizaje del alumno. El mecanismo por el que el alumno llega a realizar los cursos se basa en las votaciones y recomendaciones. Orientado a cursos basados en documentales audiovisuales y otros contenidos multimedia, el alumno, puede asignar una puntuación a los programas (contengan cursos educativos o no) que ha visto mediante la aplicación Cliente de Votación. Estas votaciones influirán en su perfil para que el Recomendador de Contenidos Educativos pueda deducir qué tipo de contenidos gustan más al alumno, puesto que en aprendizajes pasivos como los que ofrece la televisión (aunque digital aún poco interactiva) este factor cobra especial importancia a la hora de ser realizados por el usuario. De esta forma el recomendador puede generar, a petición del usuario (por ejemplo al iniciar una aplicación cliente que use los servicios del Recomendador) una conjunto de cursos recomendados para el usuario, según sus gustos: deportes, historia, idiomas, arte... El recomendador muestra al usuario una breve descripción y logotipo del curso, y ofrece un enlace para que el usuario pueda iniciar el mecanismo de despliegue (desarrollado en el proyecto OSMOSE anteriormente descrito y reutilizado por el subconsorcio español de Passepartout) y desplegar el curso (un paquete SCORM) en la plataforma. Una vez en la plataforma, el usuario puede ejecutar la aplicación tMaestro que reproducirá el curso, realizando las adaptaciones necesarias según el perfil del usuario.

El autor de esta memoria tuvo una participación activa en los siguientes problemas afrontados por los miembros españoles en Passepartout:

- Control de aplicaciones
- Recomendación y adaptación de contenidos

A continuación describimos lo más destacado en cada uno de estos puntos.

Control de aplicaciones

Para el desarrollo de la plataforma multimedia se optimizó el escritorio desarrollado para el proyecto OSMOSE, haciendo posible la interacción mediante un mando a distancia entre el usuario, el escritorio y las aplicaciones instaladas.

Para capturar la señal de un mando a distancia (se usó el mando a distancia Microsoft Media Center) se usó el paquete nativo LIRC⁸⁰ que permite capturar y enviar señales de infrarrojo de la mayoría (pero no todos) de los mandos a distancia existentes en el mercado y aplicando el patrón Envoltorio, se construyó un bundle que exportaba la funcionalidad del API de jlirc⁸¹ como servicio OSGi lo que permitió añadir soporte al escritorio para controlar las señales recibidas de los mandos.

Para ello se rediseñó el escritorio, separando las clases e interfaces básicas en un bundle independiente denominado `desktopbase`, y la implementación de dichas interfaces y clases abstractas en otro denominado `desktop`. De esta forma se favorecía la implementación de diferentes escritorios capaces de controlar aplicaciones con interfaces gráficas de usuario (descritas por medio de una instancia de la interfaz `GuiApplication` de `desktopbase`), así como la implementación de diferentes tipos de aplicaciones con diferentes tipos de interfaces gráficas de usuarios, en una forma similar a la usada por Eclipse en la construcción de plugins con interfaz gráfica para su PDE [37].

La integración del soporte para controles remoto se realizó al nivel de la especificación base para escritorios, tal y como se muestra la siguiente figura:

⁸⁰ Linux Infrared Remote Controller, <http://www.lirc.org/>

⁸¹ <http://jlirc.sourceforge.net/>

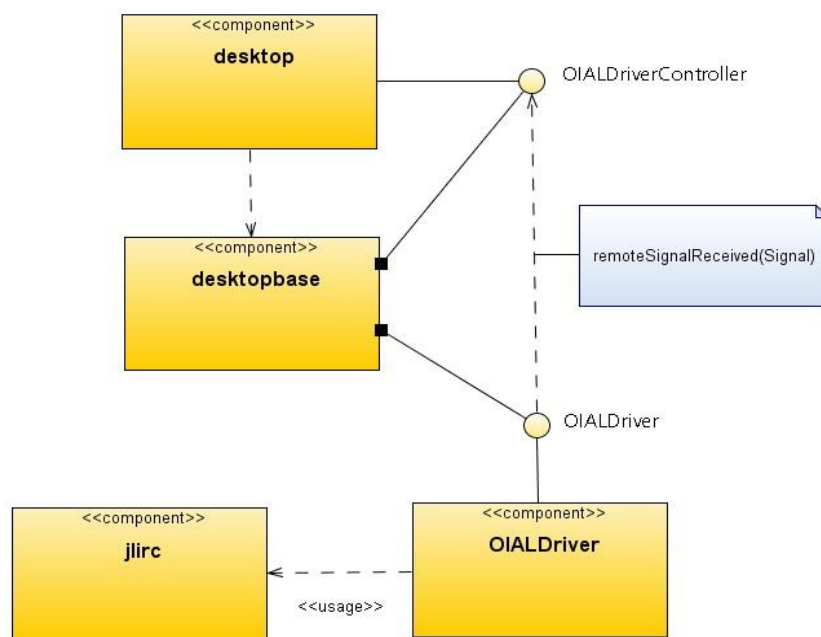


Figura 39. Soporte para control remoto por infrarrojos en el escritorio

Las interfaces `OIALDriver` y `OIALDriverController` son bastante simples y no definen ninguna lógica asociada. `OIALDriver` simplemente define un método que devuelve el nombre o identificador del driver y otro que establece el controlador encargado de recibir las señales

```

package org.gateway.home.desktopbase.service;

public interface OIALDriver {

    public String getName();

    public void setDriverController(OIALDriverController listener);

}
    
```

Por su parte la interfaz `OIALDriverController` define un conjunto de constantes con las que identificar señales asociadas a eventos de pulsación de teclas no comunes en un teclado (como por ejemplo las teclas *Play*, *Rewind*, *Forward*, *Record* o botones de colores usados para el teletexto) e igualmente un único método, `remoteSignalReceived`, que la instancia de `OIALDriver` usará para notificar la señal recibida, delegando en la instancia de `OIALDriverController`, que en nuestro caso es el propio escritorio, su procesado. Esto permite al escritorio, ante la llegada de una señal no estándar (que no podría generarse de forma inmediata con un evento de teclado del PC) diferenciar entre eventos que deben ser atendidos por él mismo o por la aplicación activa, así como la implementación de acciones de alto nivel como cerrar todas las aplicaciones activas, minimizar todas, distribuir en cascada, en mosaico, iniciar reproducción multimedia desde el CD, visualizar vídeo a pantalla completa, etc. La clase `Signal` modela una señal cuyos atributos son: `id`, una cadena con el identificador de la misma; `keyCode`, un entero que representa el código de la tecla pulsada (en caso de que la tecla pueda ser asociada con una equivalente de un PC normal); `keyChar`, el carácter alfanumérico asociado a la tecla pulsada (en caso de que se corresponda con una tecla alfanumérica); y `modifiers`, un entero que

representa los modificadores seleccionados (por ejemplo, se ha pulsado la tecla de establecer mayúsculas simultáneamente con una tecla alfanumérica).

Para eventos de bajo nivel que sí pueden generarse mediante la pulsación de una tecla en un teclado convencional de PC (es decir, pueden asociarse directamente con un evento de pulsación de una tecla definida en la clase `java.awt.event.KeyEvent`) el escritorio se limita a generar este evento mediante la clase de utilidad `Robot` del paquete `java.awt`:

```
Robot robot;
try {
    robot = new Robot();
    if (signal.getId() == OIALDriverController.VK_KP_RIGHT) {
        robot.keyPress(KeyEvent.VK_RIGHT);
        robot.keyRelease(KeyEvent.VK_RIGHT);
    }else ...
    .
    .
    .
} catch (AWTException awte) { . . . }
```

Además del problema asociado al control de aplicaciones con un mando a distancia, también influye a la hora de diseñar aplicaciones para su uso en una televisión la menor resolución de esta, que obliga al diseño de aplicaciones con mayores tamaños de letra, mayores iconos, evitar el uso del color negro (pues se convierte en transparente en una televisión) y muchos otros requisitos que aprovechamos de la experiencia previa del proyecto OSMOSE.

Recomendación y adaptación y de contenidos

Se han definido mecanismos de adaptación basados en reglas, expresadas en XML para los SCOs. De esta forma, mediante pares propiedad-valor, es posible adaptar el curso a los gustos y preferencias del usuario.

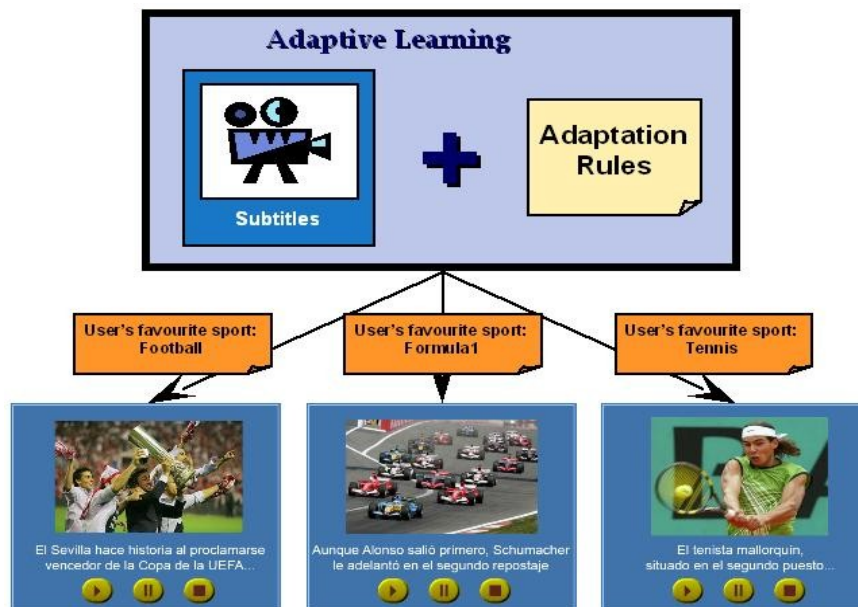


Figura 40. Esquema de adaptación de contenidos en tLearning.

La figura anterior nos informa de la adaptación respecto al deporte preferido del usuario: fútbol, automovilismo o tenis. Pero también se pueden incluir reglas de adaptación según el idioma materno del usuario, se puede modificar el tamaño de los caracteres del texto, el estilo de la música

de fondo, si quiere habilitar la función text2speech o no, etc. todo ello según las preferencias del usuario.

En el proyecto se ha obtenido una primera aproximación a la solución, basada en la adaptación de trabajos previos del LTVDI⁸² como AVATAR y tMAESTRO [38], su integración con algunos resultados del proyecto OSMOSE (gestión de despliegue de aplicaciones usando la herramienta J-Bones, mejorando el escritorio de OSMOSE para adaptar su uso a telespectadores) y el desarrollo de algunas nuevas herramientas como un Recomendador de Contenidos⁸³ usando RSS, un cliente de VLC para OSGi⁸⁴, y un Cliente de Votaciones que contribuye a anotar los perfiles de usuario en función de la valoración que ellos mismos hacen de los contenidos visualizados.

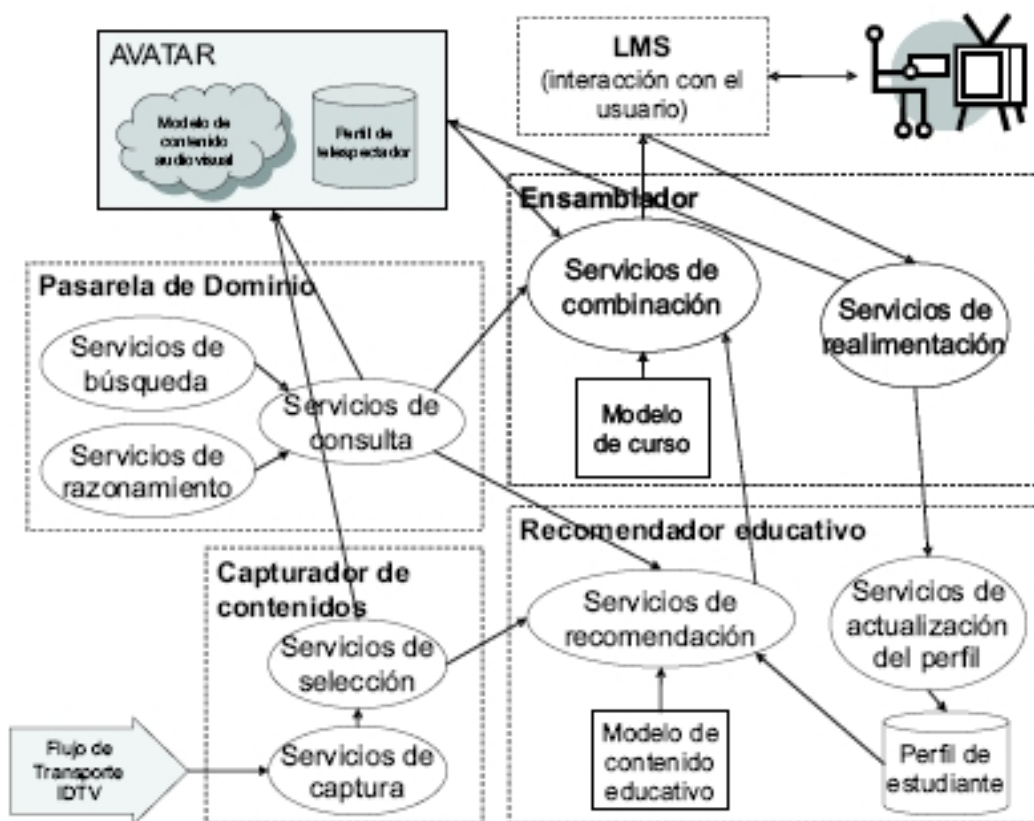


Figura 41. Módulos funcionales de tMAESTRO.

Una de las características de personalización de los cursos llevadas a cabo en Passepartout, es la inclusión de un narrador virtual, denominado avatar, que complementa los vídeos, que son el contenido educativo principal. Este avatar puede personalizarse a partir de un fichero XML en el que se describen ciertas propiedades como el paisaje (la imagen que hay tras el narrador) o la música de fondo. A través de la herramienta AVATAR desarrollada por la Universidad de Vigo se procesa este fichero XML y se crea al vuelo el avatar, que se integra en la plantilla sobre la que se visualizan los contenidos definidos en Manifiesto de SCORM (imsmanifest.xml). El problema es como adaptar esta personalización del avatar, y hacerla extensiva al resto de contenidos, para más de un usuario.

⁸² Laboratorio de Televisión Digital Interactiva. Departamento de Ingeniería Telemática. Universidad de Vigo. <http://tvdi.det.uvigo.es>

⁸³ Ofindu <http://ofindu.forge.os4os.org>

⁸⁴ VLC4OSGi <http://vlc4osgi.forge.os4os.org/>

La solución de urgencia adoptada fue crear un fichero de adaptaciones, en el que se definirían reglas para una serie de parámetros y en base al cumplimiento de una u otra regla elegir uno u otro fichero con el perfil adecuado para el avatar y el resto de propiedades para los contenidos.

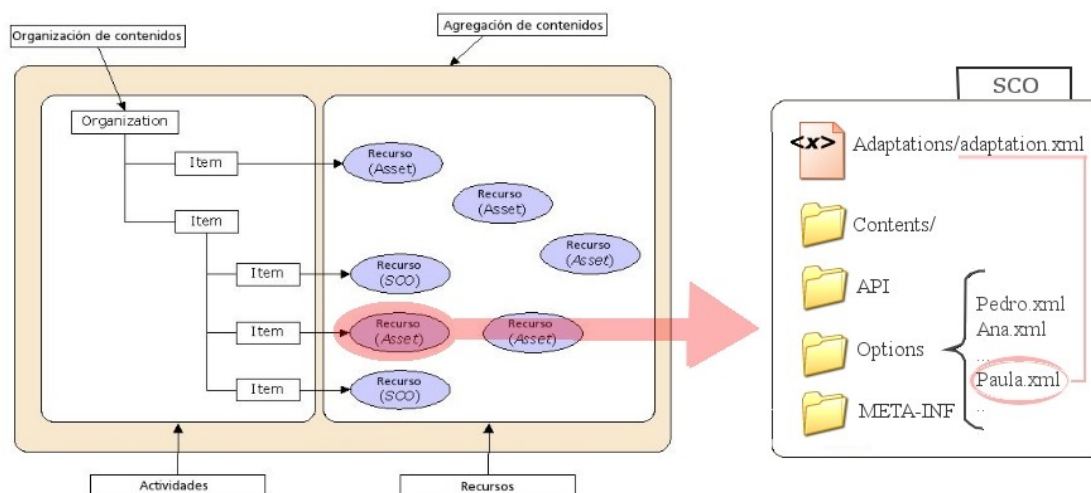


Figura 42. Inclusión del fichero de reglas de adaptación en un SCO

Las reglas de adaptación para las preferencias del usuario se implementaron al nivel de SCOs, incluyendo en los mismos un fichero XML, `adaptation.xml`, en el que se definen los parámetros a considerar y las reglas a llevar a cabo según los valores de dichos parámetros. Ejemplos de reglas de adaptación son:

- Aumentar el tamaño de la fuente y usar un fondo que proporcione mayor contraste para usuarios con ligeras deficiencias visuales.
- Cambiar el texto de los subtítulos según el idioma preferido del alumno.
- Personalización del narrador (personaliza la imagen del fondo, el narrador virtual, la y música de fondo)

En cuanto a la personalización, la aplicación tMAESTRO mostrará sólo los cursos recomendados para el usuario actual (sólo puede haber uno activo simultáneamente). Para ello usa el Recomendador de Contenidos Educativos.

En el siguiente listado se muestra un ejemplo de fichero de adaptaciones para un curso de español.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<adaptation xmlns="http://idtv.det.uvigo.es">

  <adaptationParameters>
    <adaptationParameter>
      <source>tmaestro_idtv_det_uvigo_es</source>
      <value>preferredSubject</value>
    </adaptationParameter>
    <adaptationParameter>
      <source>tmaestro_idtv_det_uvigo_es</source>
      <value>preferredMusicGenre</value>
    </adaptationParameter>
  </adaptationParameters>

  <adaptationRules>
    <rule>
      <ruleConditions>
        <ruleCondition>
          <adaptationParameter>
            <source>tmaestro_idtv_det_uvigo_es</source>
            <value>preferredSubject</value>
          </adaptationParameter>
          <comparative operator="equals">archaeology</comparative>
        </ruleCondition>
        <ruleCondition>
          <adaptationParameter>
            <source>tmaestro_idtv_det_uvigo_es</source>
            <value>preferredMusicGenre</value>
          </adaptationParameter>
          <comparative operator="equals">rock</comparative>
        </ruleCondition>
      </ruleConditions>
      <ruleAction>
        <file>Options/Pedro.xml</file>
      </ruleAction>
    </rule>

    <rule>
      <ruleConditions>
        <ruleCondition>
          <adaptationParameter>
            <source>tmaestro_idtv_det_uvigo_es</source>
            <value>preferredSubject</value>
          </adaptationParameter>
          <comparative operator="equals">arts</comparative>
        </ruleCondition>
        <ruleCondition>
          <adaptationParameter>
            <source>tmaestro_idtv_det_uvigo_es</source>
            <value>preferredMusicGenre</value>
          </adaptationParameter>
          <comparative operator="equals">orchestral</comparative>
        </ruleCondition>
      </ruleConditions>
      <ruleAction>
        <file>Options/Ana.xml</file>
      </ruleAction>
    </rule>
  </adaptationRules>
</adaptation>
```

Listado 7. Ejemplo de fichero adaptation.xml

Usando expresiones más habituales en la programación estructurada, lo que este fichero expresa es⁸⁵:

```
if ((preferredSubject == " archaeology") &&
    (preferredMusicGenre == "rock")) {
    cargarPerfil("Options/Pedro.xml");
} else if ((preferredSubject == " arts") &&
    (preferredMusicGenre == "orchestral")) {
    cargarPerfil("Options/Ana.xml");
}
```

Es decir, que si el tema preferido es arqueología y el género musical es rock, deducimos que se trata de Pedro y procedemos a cargar su perfil, para personalizar de esta forma el curso de español con imágenes de arqueología para el fondo del avatar y música rock ambiental. En cambio si el tema favorito es artes y el género musical es música de orquesta, se personalizará el avatar con un fondo en que aparece la entrada principal al Museo del Louvre y música sinfónica de ambiente. De esta forma se consigue una simple personalización de la escena del personaje que narra y hace las veces de guía del curso.

Los principales inconvenientes de esta aproximación son:

–Formato poco eficiente: se invierten muchos bytes debidos a la propia sintaxis de XML, para poca información semántica.

–Se decide el perfil a usar en función de las preferencias elegidas, cuando sería más apropiados seleccionar las preferencias en función del usuario.

–Con esta aproximación todos los perfiles se incluyen en cada uno de los SCO que componen el paquete SCORM, por lo que esta información se repite y hace menos eficiente el sistema cuanto más SCOs contenga el paquete SCORM.

–Evidentemente requiere la modificación del LMS, como así se ha realizado, para poder procesar las reglas de adaptación y presentar el curso de acuerdo a las preferencias del usuario (tamaño del texto, fondos gráfico y sonoro del avatar, etc.

Para aplicar las reglas de adaptabilidad en función del perfil del usuario que usa la plataforma sería necesario que la información de los perfiles de usuario, así como las reglas de adaptación (cuando sea posible, pues no todas las características lo permiten) no se incluyan en el SCO, sino que puedan ser gestionadas por un servicio independiente en el que el LMS se apoyará para efectuar dichas adaptaciones y que podrá ser usado también por otros servicios para ofrecer al usuario la posibilidad de modificar sus preferencias (editarlas, añadir nuevas, eliminar reglas, etc.

En las siguientes páginas se muestran algunas capturas de pantalla del sistema desarrollado, en el que se muestra un curso sobre la Unión Europea, aprovechando el 50 aniversario del Tratado de Roma:

⁸⁵ Nótese que para lenguajes orientados a objetos como Java, para comprobar si dos cadenas son la misma deberíamos usar el método equals(String) puesto que usando el operador binario == lo que comparamos son los propios objetos, que serían distintos incluso para cadenas idénticas, pero se ha optado por el uso de un posible operador binario == por motivos de claridad en la escritura.

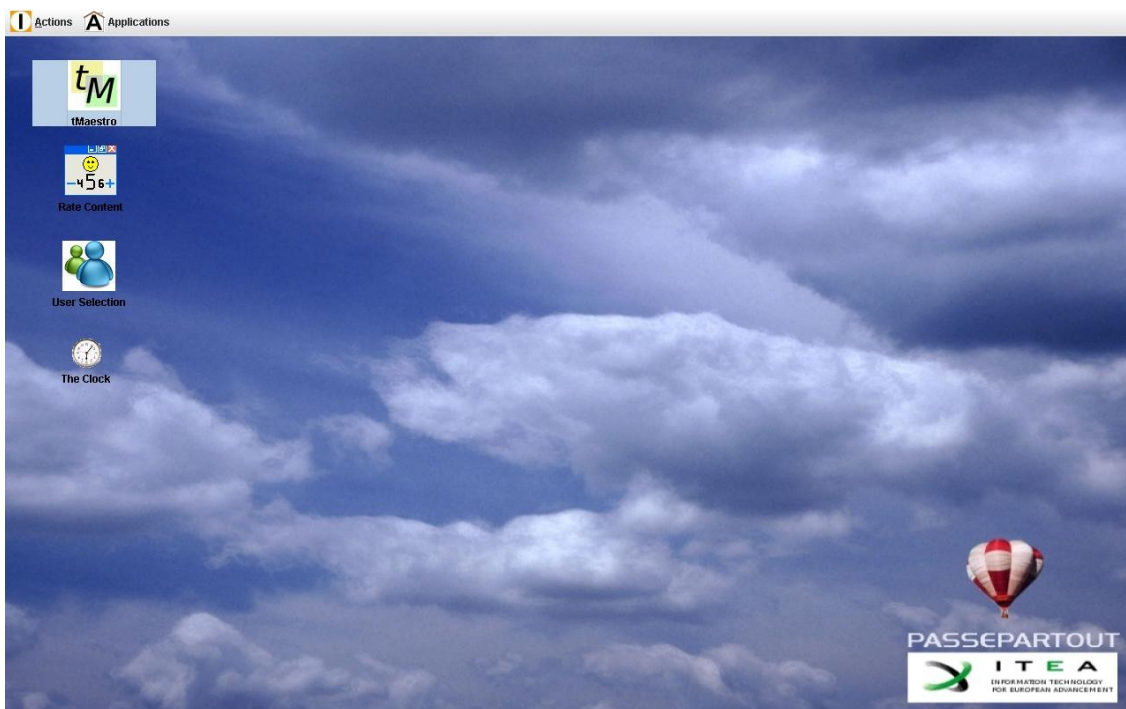


Figura 43. Escritorio: menú y accesos directos (iconos)

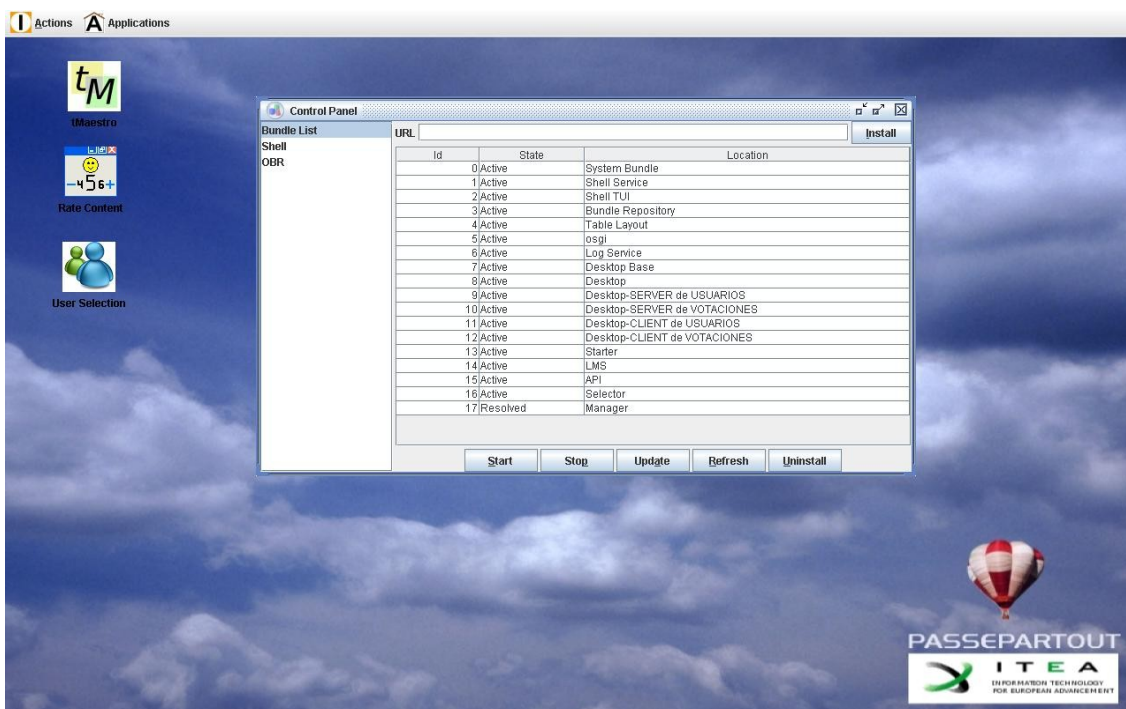


Figura 44. Control Panel: Gestor de bundles desplegados en la plataforma.

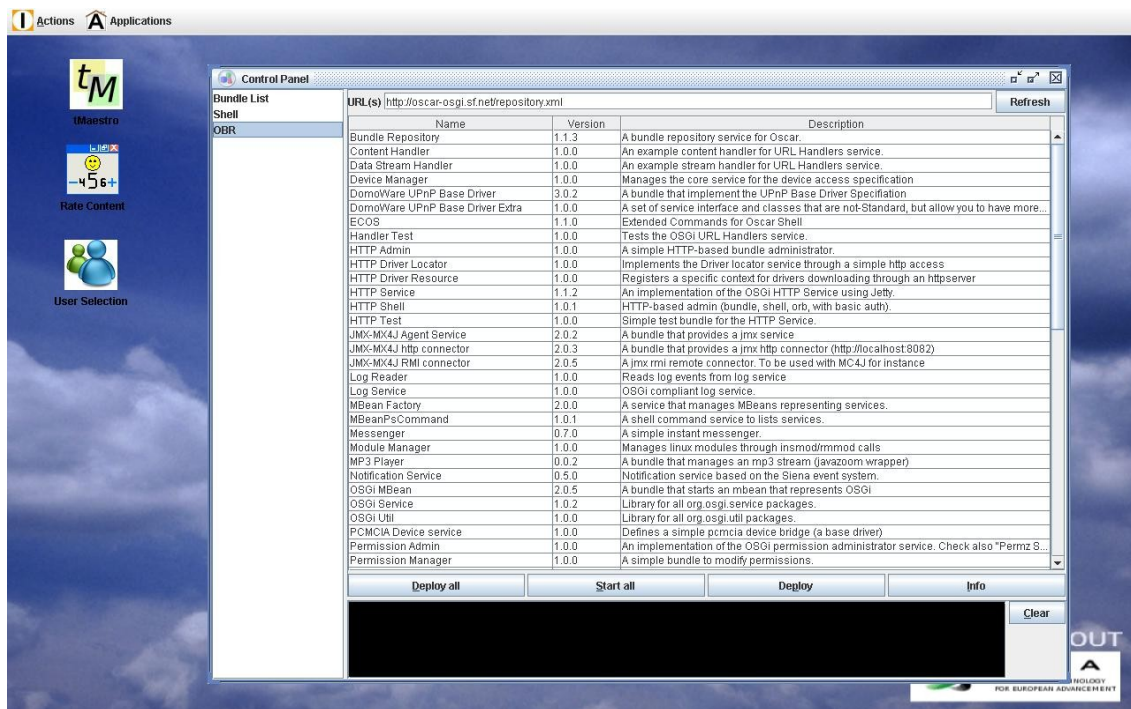


Figura 45. Control Panel: Acceso al Oscar Bundle Repository para obtener nuevas utilidades.

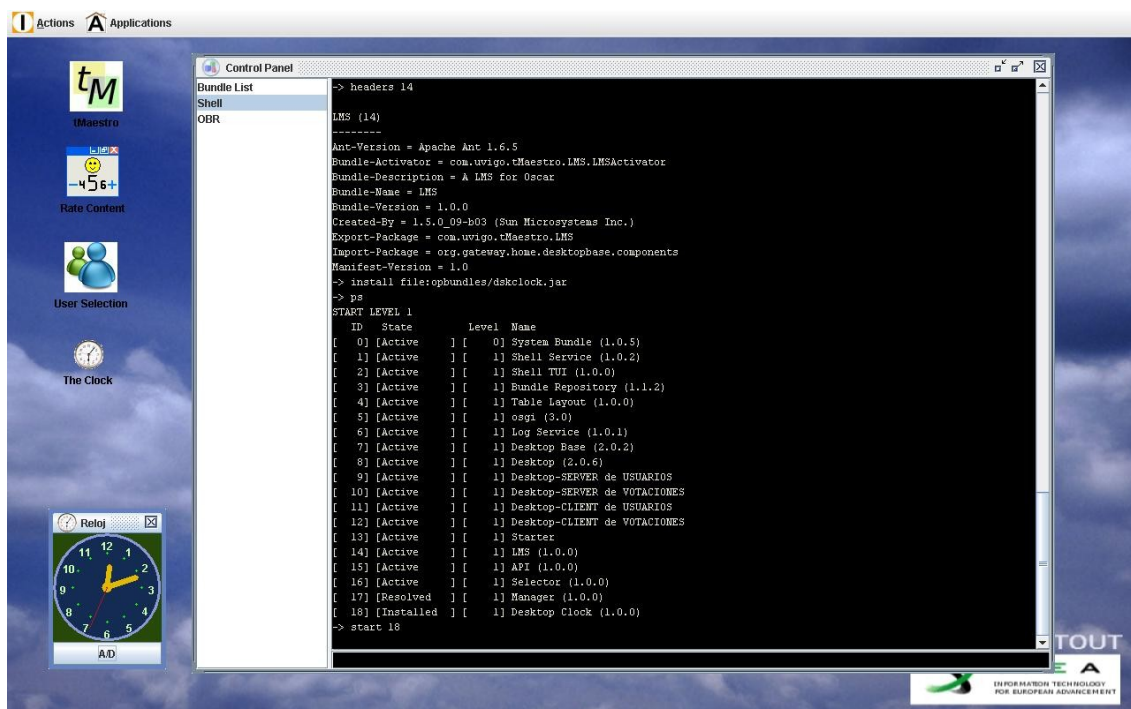


Figura 46. Control Panel⁸⁶: Utilidad Shell que permite ejecutar comandos del framework.

⁸⁶ El reloj que aparece en la figura se ha adaptado fácilmente del proyecto Eduka.

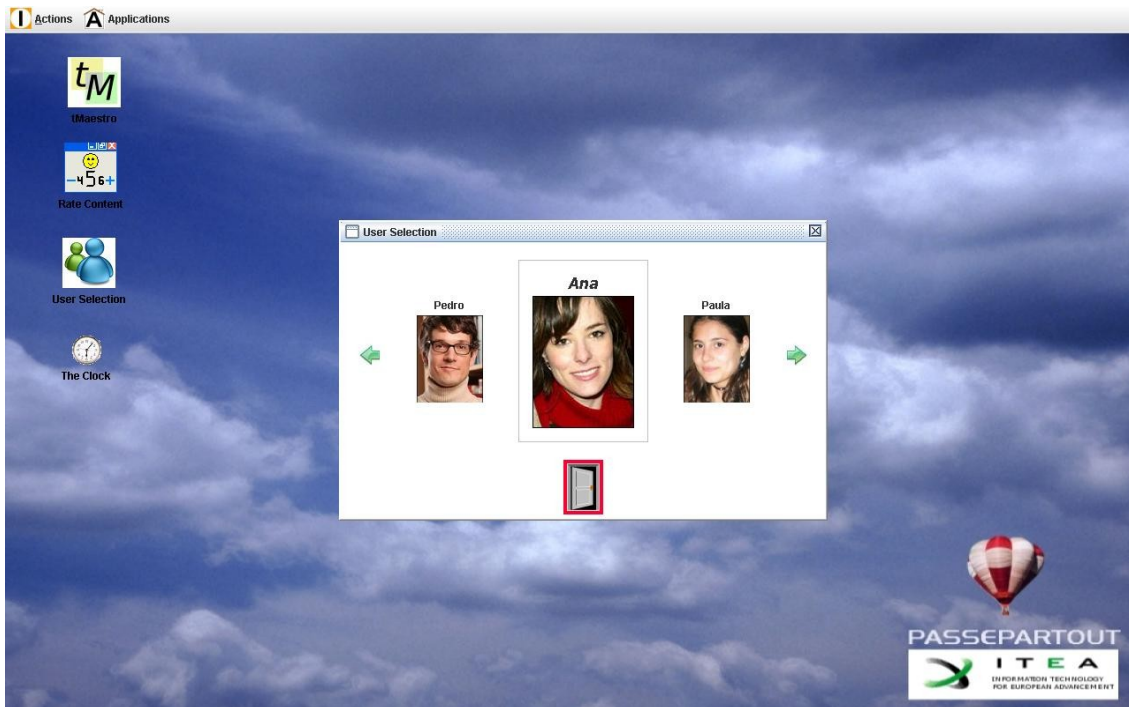


Figura 47. Selección de usuario (Ana)

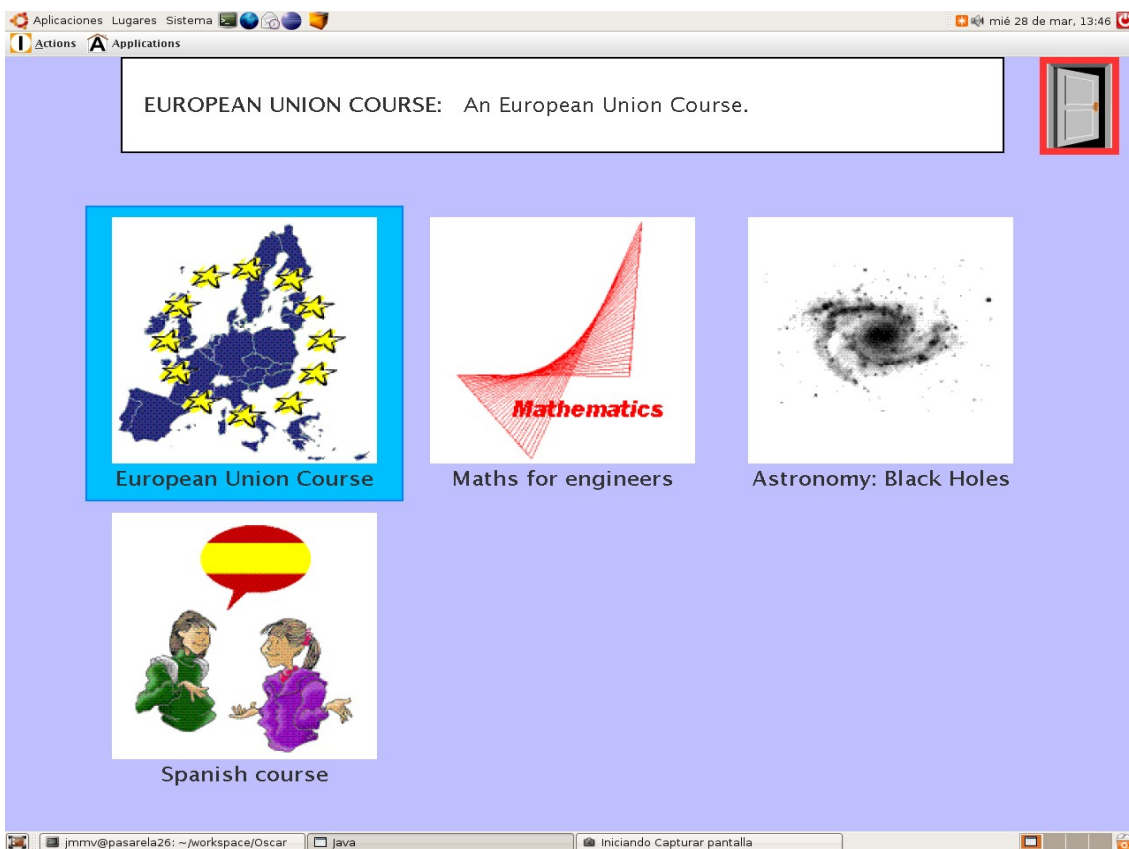


Figura 48. Recomendación de cursos para Ana en la aplicación tMAESTRO.

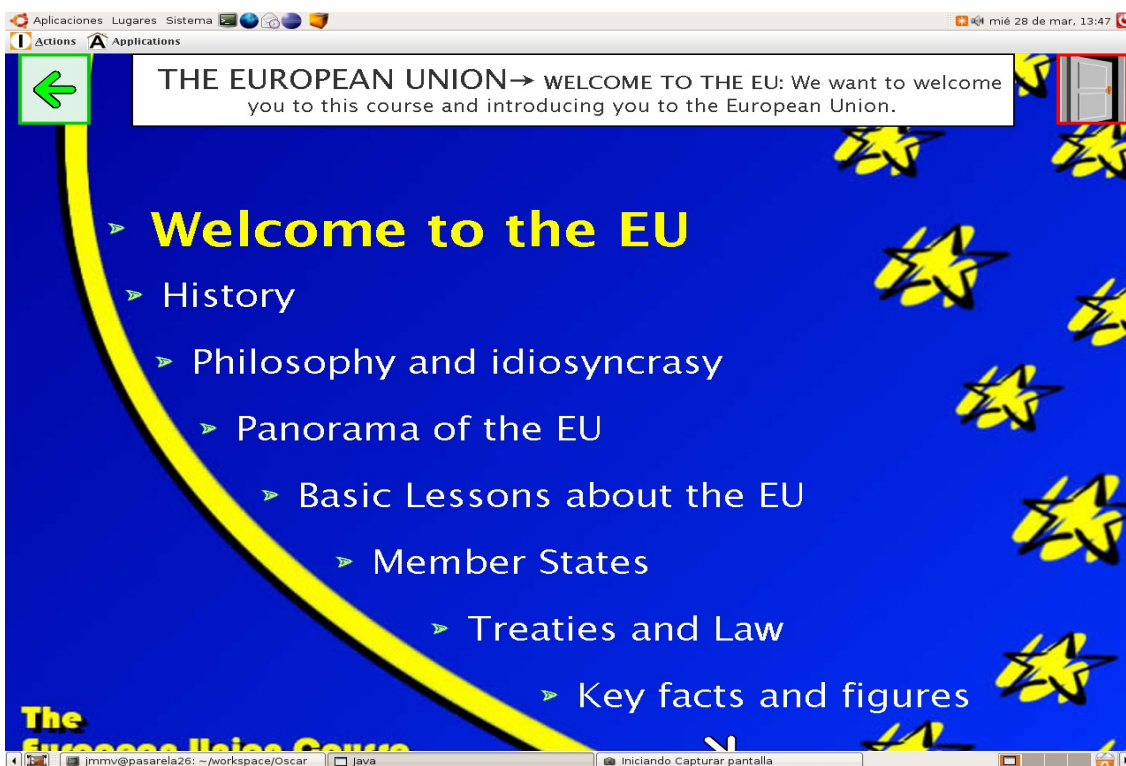


Figura 49. Primera pantalla del curso sobre la Unión Europea



Figura 50. Navegación con el control remoto en el guión del curso SCORM



Figura 51. Selección del primer punto del primer capítulo: History – A little journey...



Figura 52. Visualización del recurso asociado al SCO (asset) seleccionado (I).



Figura 53. Visualización del recurso asociado al SCO (asset) seleccionado (II).

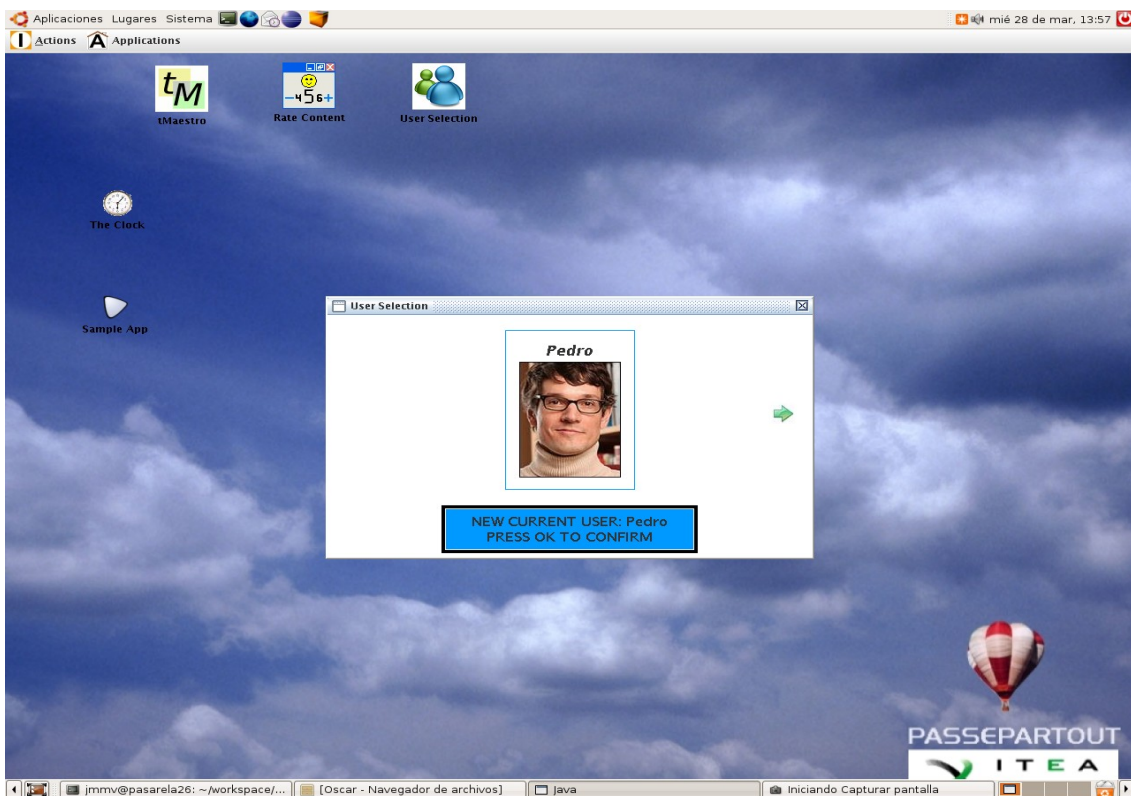


Figura 54. Selección de usuario (Pedro)

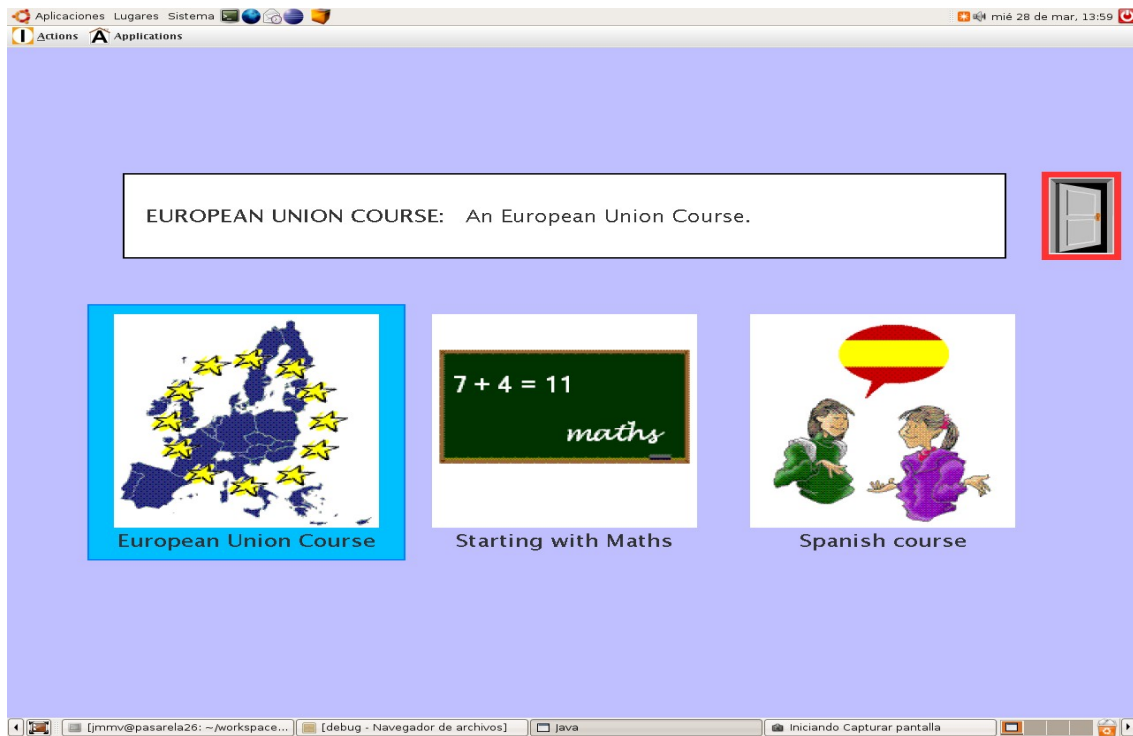


Figura 55. Recomendación de cursos para Pedro en la aplicación tMAESTRO.



Figura 56. Visualización del recurso asociado al SCO.

Obsérvese la diferente personalización y adaptación del texto para su deficiencia visual.

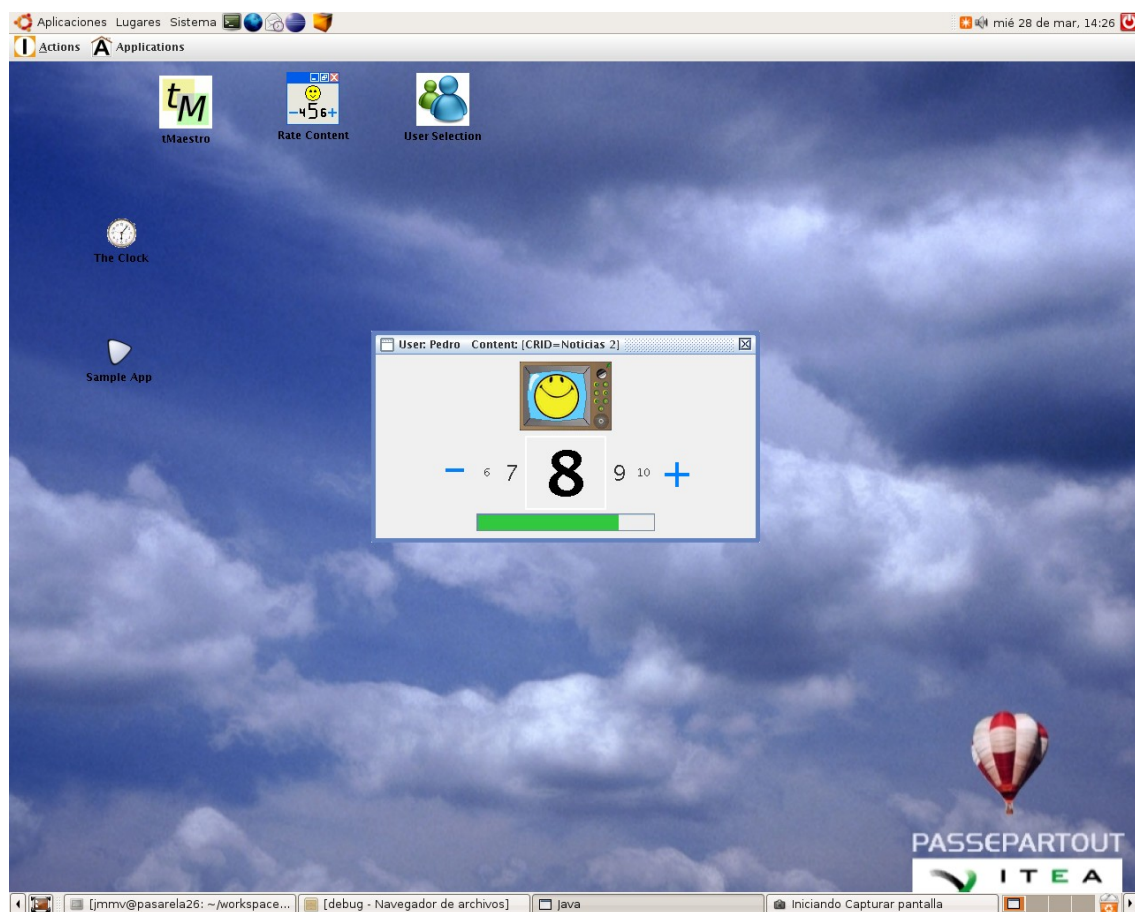


Figura 57. Votación de contenidos: Pedro otorga un 8 a "Telediario 2".

5. Conclusiones

En este trabajo de investigación se ha mostrado un análisis-resumen, enumerado algunos problemas relacionados con las especificaciones en eLearning:

-El inconveniente que supone para algunos dispositivos la obligación de usar javascript en SCORM. Esto supone un problema en el mercado de dispositivos móviles como teléfonos y PDAs de gama baja que o bien no cuentan ni siquiera con un navegador web o bien a pesar de contar con él, éste no cuenta con un motor de javascript.

-SCORM no define explícitamente dónde debe ir el LMS, pudiendo ir en el cliente o en el servidor. La mayoría de sistemas de eLearning actuales optan por ubicar el gestor de aprendizaje en el servidor, debiendo, ante los eventos generados (superación de una unidad didáctica, petición de realizar un test, etc.) comunicar el curso con el LMS remoto haciendo uso del API javascript que se incluye en el propio paquete del curso. En muchos casos, esto introduce dos inconvenientes: primero, el curso no puede ser seguido en su totalidad sin conexión a Internet, y segundo, este mecanismo dificulta la implementación de un sistema que permita suplantar al LMS remoto por uno local encargado de guardar los eventos producidos para posteriormente, al volver al modo on-line, sincronizar con el LMS remoto enviándole estos eventos. Para ello es necesario poder asociar cada curso con el LMS remoto (en el servidor) encargado de llevar el seguimiento de dicho curso.

-La dificultad para expresar formalmente la especificación de IMS Learning Design

-La inexistencia de especificaciones que permitan el uso de técnicas de adaptación de los contenidos (según las características del dispositivo, del contexto, del usuario...), las dificultades y, al mismo tiempo, la libertad a la hora de implementar el mecanismo adecuado para llevar a cabo estas adaptaciones.

-La arquitectura de la mayoría de LMS existentes no facilita la adición de nuevos módulos.

Hemos apuntado algunas soluciones que ayudarían a minimizar algunos de estos problemas:

-Una arquitectura orientada a servicios, y en especial OSGi, ayudaría a conseguir una arquitectura modular, orientada a servicios y que permitiría la incorporación de nuevos módulos en tiempo de ejecución.

-Se han identificado similitudes entre paquetes IMS CP y bundles OSGi, y se ha probado en un proyecto que es posible distribuir y procesar los cursos SCORM como bundles.

-Se ha implementado una primera aproximación de plataforma, pensada para ser ubicada en una pasarela residencial, integrando TV digital y aplicaciones Java, MHP y Swing.

-Se ha implementado un escritorio para esta plataforma, y se ha definido un mecanismo mediante el cual otros componentes pueden registrar sus aplicaciones para que sean usadas y controladas por dicho escritorio. Se basa en la definición de un API y un servicio OSGi, aunque sería deseable poder hacerse declarativamente (por ejemplo mediante un descriptor de la aplicación en XML).

-Se ha adaptado un reproductor de cursos al escritorio definido para la plataforma OSGi.

-Se ha implementado una primera aproximación a un mecanismo de adaptación de los contenidos, basado en la definición de reglas y acciones en XML.

Otro tipo de adaptaciones afecta al modelo de interacción y navegación por los contenidos. Los dispositivos físicos utilizados por el usuario para interactuar con el sistema influyen en la interfaz gráfica de éste. Mas bien, la interfaz gráfica no debe diseñarse únicamente pensando en la resolución máxima que podrá utilizar el cliente, sino que también ha de diseñarse la organización de ésta (elementos gráficos y tipos de eventos) teniendo en cuenta qué dispositivo(s) se usará para interactuar con ella: un ratón, un mando a distancia, pantalla táctil...

5.1. Trabajo futuro

Para un trabajo futuro se planea abordar los siguientes problemas:

-Definir el API que permita a los cursos comunicar con el LMS local y la forma en que éste comunicará con el LMS remoto, independientemente de su tecnología.

-Investigar más en como clasificar las reglas de adaptación, en función de su dependencia con respecto a la plataforma, el entorno de ejecución, el usuario, los contenidos, etc. para ver cuáles de ellas pueden ser externas al paquete de contenidos (es decir común a todos los cursos) y cuáles deben incluirse en los propios cursos.

-Estudiar cuál es el sitio idóneo para incluir las reglas de transformación (pensamos que sería adecuado incluir un apartado <adaptation>...</adaptation> como una posible organización más en el manifiesto del paquete IMS CP)

-Estudiar los requisitos en navegación y adaptabilidad para pequeños dispositivos.

-Investigar la aplicación del sistema definido para orientar el aprendizaje en base a las competencias necesarias para un determinado puesto (por ejemplo, en una empresa se requiere para un puesto determinadas competencias) Si un empleado tiene un subconjunto de ellas, ¿qué acciones formativas debe llevar a cabo para adquirir las restantes y poder optar al puesto?

6. Curriculum Vitae

José Manuel Márquez Vázquez (Sevilla, 1976) es Ingeniero en Informática (E.T.S. de Ingeniería Informática de Sevilla, 2002)

En 2007 obtuvo el Certificado de Aptitud Pedagógica (Instituto de Ciencias de la Educación de Sevilla)

En 2006 fue nombrado Asistente Honorario del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla para el curso académico 2006/2007.

En 2003 recibió el premio Mejor Investigación Docente 2001/2002 de la Universidad de Sevilla como colaborador en el proyecto “Aplicación de Sistemas Hipermedia al desarrollo de clases virtuales y semipresenciales” dirigido por Juan Antonio Ortega Ramírez. También en 2003 le fue concedido el premio “Mejor Proyecto Fin de Carrera 2003” patrocinado por FIDETIA, por el proyecto “Eduka, una primera aproximación a un sistema de educación semipresencial”.

Desde mayo de 2001 comienza a trabajar para Telvent, formando parte del Departamento de Investigación y Desarrollo desde noviembre de 2003. Como docente impartió en el I.E.S. Los Viveros (Sevilla) el curso de formación profesional ocupacional “Programador de aplicaciones de Redes Internet”, de una duración de 380 horas (del 02-12-2002 al 28-03-2003)

En relación con el tema de esta memoria es coautor de las siguientes publicaciones, que se muestran en orden cronológico:

–**Aplicación de las nuevas tecnologías: docencia interactiva.** J. A. Ortega, J. Torres, J. M. Márquez. Revista de Enseñanza Universitaria. Diciembre 2001, Número 18. Universidad de Sevilla. (Sevilla, 2001) ISSN: 1131-5245

–**Aplicación de Sistemas Hipermedia al desarrollo de clases virtuales y semipresenciales.** J. A. Ortega, J. Torres, J. M. Márquez. Innovaciones Docentes en la Universidad de Sevilla. Curso 2001-2002. Colección “Innovación y Desarrollo de la calidad de la enseñanza universitaria” N° 4. (Sevilla, 2002) ISBN: 84-86849-29-2. Recibió el Premio a la Innovación Docente en la Universidad de Sevilla para el Curso 2001/2002, concedido por el Instituto de Ciencias de la Educación el 24 de Marzo de 2003.

–**Representación Cualitativa del Conocimiento: aplicación a la generación automática de itinerarios culturales.** J. A. Ortega, J. Torres, R. M. Gasca, M. Mejías, J. M. Márquez. IV Jornadas de ARCA, JARCA'02. (Vilanova i la Geltrú, 2002) ISBN: 84-95499-62-2

–**Generación automática de itinerarios culturales aplicando EJB y programación con restricciones.** J.A. Ortega, F.Ruz, M.J. Escalona, M. Mejías, J.M. Márquez. Actas de las III Jornadas de Trabajo Dolmen. Editado por Universidad de Murcia. (El Escorial, 2002)

–**Desacoplado clases conceptuales de clases navegacionales.** A. M. Reina, J. Torres, M. Toro, M. J. Escalona, J. M. Márquez. Actas de las III Jornadas de Trabajo Dolmen. Editado por Universidad de Murcia. (El Escorial, 2002)

–**Patrones de diseño en un Sistema Multimedia para la Educación.** J. A. Ortega, J. M. Márquez, J. Torres, M. J. Escalona. Actas de las III Jornadas de Trabajo Dolmen. Editado por Universidad de Murcia. (El Escorial, 2002)

–**Definición y representación de reglas de transformación de un modelo independiente en otro específico de la plataforma.** J. A. Ortega, J. M. Márquez, A. Reina, J. A. Álvarez. Actas de las IV Jornadas de Trabajo Dolmen. Editado por la Universidad de Granada. (Alicante, 2003).

–**Catálogo de aplicaciones de OSMOSE: Una experiencia real en el uso de servicios web en plataformas OSGi.** José Manuel Márquez, Juan Antonio Ortega, Guillermo Hernández. Actas de las I Jornadas Científico-Técnicas en Servicios Web, JSWEB'2005. I Congreso Español de Informática, CEDI 2005. Editado por Thomson. (Granada, del 13 al 16 de septiembre de 2005). ISBN: 84-9732-455-2

–**Sistema avanzado de despliegue de aplicaciones en pasarelas residenciales.** Miguel García Longarón, Fernando Usero Fuentes, José Manuel Márquez, Miguel Ángel Oltra Rodríguez, José Luis Ruiz Revuelta y Juan Carlos Dueñas López. XV Jornadas Telecom I+D (Madrid, del 22 al 24 de noviembre de 2005). ISBN: 84-689-3794-0

Ha participado en los siguientes proyectos de I+D+i:

Título	JULES VERNE ⁸⁷
Código	ITEA 02002
Empresa	Telvent Interactiva, S.A.
Duración	2 años. (2003 y 2004)
Coordinador	Keith Baker {keith.baker@philips.com}
Financiación	376.158,50 € (suma total de ayudas concedidas para 2003 ⁸² y 2004 ⁸³)
Descripción	Proyecto de investigación en el que participan las mayores empresas de la industria de difusión digital interactiva para realizar pruebas e investigar las tecnologías disponibles para la creación de contenidos y capacidad de los futuros terminales y redes domésticas.

Título	OSMOSE (Open Source Middleware for Open Systems in Europe)
Código	ITEA 02003
Empresa	Telvent Interactiva, S.A.
Duración	2 años. (Junio de 2003 a Junio de 2005)
Coordinador	Jesús Bermejo. {jesus.bermejo@telvent.abengoa.com}
Financiación	1.089.232,70 € (total de ayudas concedidas para 2003 ⁸⁸ , 2004 ⁸⁹ y 2005 ⁹⁰)
Descripción	Definir un conjunto de elementos hardware y software de código libre, que facilitase la comunicación entre dispositivos en el entorno residencial, así como que garantizase, en la medida de lo posible, la ejecución de aplicaciones multimedia y su integración con la TV digital.

⁸⁷ <http://www.citi.tudor.lu/julesverne>

⁸⁸ <http://www.mecd.es/ciencia/fomento/files/PNTIC.pdf>

⁸⁹ <http://www.boe.es/boe/dias/2005/04/08/pdfs/A12269-12296.pdf>

⁹⁰ http://sise.fecyt.es/.../20_Plan%202004-

[2007/20_2005/00_Md/00_Proj.%20I%20mas%20D/01_DGDSI/POR%20TITULO.xls](http://sise.fecyt.es/.../2007/20_2005/00_Md/00_Proj.%20I%20mas%20D/01_DGDSI/POR%20TITULO.xls)

Título	FAMILIES ⁹¹ (Madurez basada en hechos mediante lecciones aprendidas en la institucionalización y la exploración asociada a la ingeniería de familias de sistemas)
Código	ITEA 02009
Empresa	Telvent Interactiva, S.A.
Duración	2 años. (Junio de 2003 a Junio de 2005)
Coordinador	Dr. Frank van der Linden {frank.van.der.linden@philips.com}
Financiación	574.797,95 € (suma total de ayudas concedidas para 2003 ⁹² , 2004 ⁹³ y 2005 ⁹⁴)
Descripción	Ingeniería del software aplicada a las familias de sistemas y productos.

Título	Passepartout
Código	ITEA 04017
Empresa	Telvent Interactiva, S.A.
Duración	2 años. (Marzo de 2005 a Marzo de 2007)
Coordinador	Keith Baker {keith.baker@philips.com}
Financiación	450.285,00 € (ayudas concedidas en la convocatoria de 2005 ⁹⁴)
Descripción	Este proyecto se centra en la convergencia de sistemas y aplicaciones digitales en media-centers en el hogar. Se espera que de este proyecto surjan nuevas tecnologías que impulsen a las industrias del software europeas hacia la convergencia sobre terminales y red con el objetivo final de la "inteligencia ambiental". El proyecto persigue aunar los media-centers con las redes domésticas para proporcionar contenidos escalables desde la televisión de alta definición (HDTV) a definiciones inferiores de una forma adecuada.

Título	OSIRIS ⁹²
Código	ITEA 04040
Empresa	Telvent Interactiva, S.A.
Duración	3 años. (2006, 2007 y 2008)
Coordinador	Jesús Bermejo {jesus.bermejo@telvent.abengoa.com}
Financiación	501.942,70 € (total de ayudas concedidas en la convocatoria de 2005 ⁹⁴)
Descripción	Integración de tecnologías heterogéneas en tiempo de ejecución para proporcionar una plataforma de servicios, abierta y escalable, en la que puedan llevarse a cabo el despliegue, agregación, distribución y adaptación dinámica de servicios al contexto en el que dicha plataforma se encuentra, proporcionando mecanismos de descubrimiento e interacción con otros servicios ubicados en diferentes nodos de la red de nodos OSIRIS.

⁹¹ <http://www.esi.es/Families/>

⁹² <http://www.itea-osiris.org>

7. Referencias bibliográficas

- [1] Gabriel Kaplún (2005). *Aprender y enseñar en tiempos de internet. Formación profesional a distancia y nuevas tecnologías*. Montevideo. CINTERFOR/OIT, 2005.
- [2] A. Armellini, J. Grunberg (2001). *El uso educativo de Internet. Conceptos e implementación*. Ponencia en Jornadas ISTECS, Montevideo, ISTECS-UdelaR-ORT-UNESCO, 2001.
- [3] empírica: LearnInd 2006 (CTS). European Commission. Information Society and Media Directorate General. *Benchmarking Access and Use of ICT in European Schools 2006. Final Report from Head Teacher and Classroom Teacher Surveys in 27 European Countries*. http://ec.europa.eu/information_society/eeurope/i2010/docs/studies/final_report_3.pdf
Publicado en: http://ec.europa.eu/information_society/eeurope/i2010/benchmarking/index_en.htm. Página accedida el 28 de diciembre de 2006.
- [4] Francesc Pedro, (2006). *Proyecto campus digital. Informe de consultoría*. Montevideo: UdelaR. Mimeo. 2006
- [5] M. Castells, (1997). *La era de la información*. Madrid. Alianza Editorial. 1997
- [6] Cristóbal Torres Albero, (2002). *El Impacto de las Nuevas Tecnologías en la Educación Superior: un Enfoque Sociológico*. Boletín de la Red Estatal de Docencia Universitaria. Vol 2. Nº 3. 2002
- [7] Ministerio de Educación y Ciencia (1989). *Libro blanco para la reforma del sistema educativo*. Madrid. MEC. 1989
- [8] J. Salinas (2000). *El rol del profesorado en el mundo digital*. Ed. L, del Carmén, pp. 305-320. Simposio sobre la formación inicial de los profesionales de la educación. Universidad de Girona. 2000.
- [9] Julio Cabero y otros, (2003). *Las nuevas tecnologías en la actividad universitaria*. Pixel-bit. Revista de Medios y Educación. Nº 20. Enero 2003, pp. 81-100.
- [10] M. Fernández Enguita, (1990). *Las funciones sociales de la escuela*. En: La escuela a examen. Madrid. Ed. Eudema. 1990
- [11] Red.es, (2006). Observatorio de las Telecomunicaciones y la Sociedad de la Información. *Perfil sociodemográfico de los internautas. Análisis de datos INE*. Septiembre de 2006. http://observatorio.red.es/estudios/documentos/magnitudes_sociodemograficas_sep.pdf
Publicado en <http://observatorio.red.es/> Último acceso: 5 de febrero de 2007.
- [12] Juliana Correa, Teresa Paniagua, (1997). *Capítulo III. Contenidos y Metodología*. De: C.A.P. Materiales Didácticos. Varios autores. Editado por ICE Universidad de Sevilla, 1997.
- [13] María José Auñón, (2000). Equipos de producción de cursos a distancia. Barcelona. Universidad de Barcelona. MIMEO. 2000
- [14] Hilary Perraton, Charlotte Creed, Bernadette Robinson, (2002) *Teacher education guidelines: using open and distance learning*. UNESCO, 2002.
- [15] Pedraglio, G. (2002). *Une exception française*. Centraliens, num. 535, p.-1. Paris. France. 2002

- [16] Khan, B.H. (2005) *Learning features in an open, flexible, and distributed environment*. AACE Journal, 13(2), 137-153. 2005.
- [17] David Griffiths, Josep Blat, Rocío García, Sergio Sayago, (2004). *La aportación de IMS Learning Design a la creación de recursos pedagógicos reutilizables*. SPDCE 2004. I Simposio Pluridisciplinar sobre Diseño, Evaluación y Descripción de Contenidos Educativos Reutilizables. 20-22 Octubre 2004. Universidad de Alcalá, Guadalajara. España.
- [18] Tattersal et al. (2003). *IMS Learning Design Frequently Asked Question*. 2003.
- [19] R. Koper, B. Olivier, (2004). *Representing the Learning Design of Units of Learning*. Educational Technology & Society, 7 (3), 97-111. 2004
- [20] Badrul H. Khan, (2001). *A Framework for Open, Flexible and Distributed ELearning*. <http://BooksToRead.com/framework> 2001
- [21] Ricardo R. Amorim, Eduardo Sánchez, Manuel Lama, (2005). La utilización de una ontología IMS-LD en el modelado de diseños de aprendizaje. XI Conferencia de la Asociación Española para la Inteligencia Artificial CAEPIA 2005. Santiago de Compostela. España. 2005.
- [22] J. Dalziel, (2004). *Open standards versus open source in e-learning*. EDUCAUSE Quarterly, 4. 2004. <http://www.educause.edu/ir/library/pdf/EQM0340.pdf>
- [23] Xiaofei Liu, Abdulmotaleb El Saddik, Nicolas D. Georganas, (2003). *An implementable architecture of an e-learning system*. CCECE 2003 – CCGEI 2003, Montreal, 2003
- [24] Sun Microsystems, (2003). *E-Learning Framework. Technical White Paper*. <http://www.sun.com/products-n-solutions/edu/whitepapers/pdf/framework.pdf>
- [25] Daniela Leal Musa, José Palazzo Moreira de Oliveira, (2004). *Sharing Learner Information through a Web Services-based Learning Architecture*. International Workshop on Web Information Systems Modeling, WISM 2004.
- [26] Comisión de las Comunidades Europeas, (2006). *Propuesta de Recomendación del Parlamento Europeo y del Consejo relativa a la creación del Marco Europeo de Cualificaciones para el aprendizaje permanente*. Bruselas, 05.09.2006. http://ec.europa.eu/education/policies/educ/eqf/com_2006_0479_es.pdf
- [27] ISO/IEC 9126: The Standard of Reference. (1991) <http://www.cse.dcu.ie/essiscope/sm2/9126ref.html>
- [28] J. Massey, Quality and e-learning in Europe, (2003). *Lifelong Learning: a citizens' views*. Office for official publication of the European Communities. Luxembourg, 2003
- [29] B. Schilit, N. Adams, and R. Want. (1994). *Context-aware computing applications*. IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94), Santa Cruz, CA, US: 89-101.
- [30] Guanling Chen , David Kotz, (2000). *A Survey of Context-Aware Mobile Computing*. Research, Dartmouth College, Hanover, NH, 2000
- [31] J. C. Dueñas, J. L. Ruiz, J. Bermejo, J. A. Alonso, C. Acuña, C. Díaz. (2004) "Plataformas abiertas para la provisión de servicios", XIV JORNADAS TELECOM I+D. Madrid 23,24,25 Noviembre 2004.
- [32] A. Saad. (2003) *Java-based Functionality and Data Management in the Automobile. Prototyping at BMW Car IT GmbH*. Java Spectrum, March 2003.
- [33] Sun Microsystems Inc. (1999) JAR File Specification. <http://java.sun.com/j2se/1.3/docs/guide/jar/jar.html>
- [34] APT-GET, <http://www.appt-get.org/>

- [35] Miguel García Longarón, Fernando Usero Fuentes, José Manuel Márquez, Miguel Ángel Oltra Rodríguez, José Luis Ruiz Revuelta y Juan Carlos Dueñas López, (2005). *Sistema avanzado de despliegue de aplicaciones en pasarelas residenciales*. XV Jornadas Telecom I+D (Madrid, del 22 al 24 de noviembre de 2005)
- [36] José Manuel Márquez, Juan Antonio Ortega, Guillermo Hernández (2005). *Catálogo de aplicaciones de OSGi: Una experiencia real en el uso de servicios web en plataformas OSGi*. Actas de las I Jornadas Científico-Técnicas en Servicios Web, JSWEB'2005. I Congreso Español de Informática, CEDI 2005.
- [37] Jeff McAffer, Jean-Michel Lemieux. (2005). *Eclipse Rich Client Platform: Designing, coding, and Packaging Java Applications*. The eclipse series. Addison-Wesley. 2005.
- [38] M. Rey López, R. P. Díaz Redondo, A. Fernández Vilas (2005). *Educación a la carta para IDTV*. V Jornadas de Ingeniería Telemática JITEL 2005. Vigo, 12-14 septiembre de 2005.
- [39] Mercedes Fernández García, Manuel Fernández González (2003). *Ensayo de plataforma para eLearning en las disciplinas de Ingeniería*. Jornada de innovación y calidad docente en la docencia universitaria. El espacio europeo de educación superior. Universidad Pontificia de Comillas. Instituto de Ciencias de la Educación. Madrid, 2003.
- [40] J-Bones, <http://jbones.forge.os4os.org/>
- [41] J. C. Dueñas, J. L. Ruiz, and M. Santillán. *An End-to-End Service Provisioning Scenario for the Residential Environment*. IEEE Communications Magazine, vol. 43, no. 9, Septiembre 2005.
- [42] A. M. Reina, J. Torres, M. Toro, M. J. Escalona, J. M. Márquez. *Desacoplando clases conceptuales de clases navegacionales*. Actas de las III Jornadas de Trabajo Dolmen. Editado por Universidad de Murcia. (El Escorial, 2002)
- [43] J. A. Ortega, J. Torres, J. M. Márquez. *Aplicación de las nuevas tecnologías: docencia interactiva*. Revista de Enseñanza Universitaria. Diciembre 2001, Número 18. Universidad de Sevilla. (Sevilla, 2001) ISSN: 1131-5245
- [44] J. A. Ortega, J. Torres, J. M. Márquez. *Aplicación de Sistemas Hipermedia al desarrollo de clases virtuales y semipresenciales*. Innovaciones Docentes en la Universidad de Sevilla. Curso 2001-2002. Colección "Innovación y Desarrollo de la calidad de la enseñanza universitaria" N°4. (Sevilla, 2002) ISBN: 84-86849-29-2
- [45] J. A. Ortega, J. M. Márquez, J. Torres, M. J. Escalona. *Patrones de diseño en un Sistema Multimedia para la Educación*. Actas de las III Jornadas de Trabajo Dolmen. Editado por Universidad de Murcia. (El Escorial, 2002)
- [46] Tesis Doctoral, A POLICY-DRIVEN, MODEL-BASED SOFTWARE AND SERVICES DEPLOYMENT ARCHITECTURE FOR HETEROGENEOUS ENVIRONMENTS. Jose L. Ruiz, dirigido por Juan C. Dueñas, Universidad Politécnica de Madrid, por el Departamento de Ingeniería de Sistemas Telemáticos en periodo de exposición pública, Jun 2007.
- [47] J2EE, <http://java.sun.com/j2ee/index.jsp>
- [48] Apache Axis, <http://ws.apache.org/axis/>
- [49] OSGi Application Manager, <http://forge.os4os.org/projects/tid-oam/>
- [50] Web Services Support for OSGi: The Axis Bundle.
<http://forge.os4os.org/projects/ws-support/>

8. Foros relacionados

A continuación se citan algunos foros y revistas relacionados con las áreas involucradas en el presente trabajo:

8.1. Foros y portales en internet

1. UNESCO

<http://www.unesco.org>

2. ELearning Workshops

<http://www.elearningworkshops.com>

3. Elearningeuropa.info. Portal de eLearning financiado por la Comisión Europea.

<http://www.elearningeuropa.info/>

4. The M-Learning Project.

<http://www.m-learning.org/>

5. M-Learning Forum.

<http://www.pjb.co.uk/m-learning/>

6. The T-Learning study.

<http://www.pjb.co.uk/t-learning/>

7. Multimedia Home Platform. (MHP)

<http://www.mhp.org/>

8. World Wide Web Consortium: Synchronized Multimedia

<http://www.w3c.org/AudioVideo>

9. Televisión Digital Interactiva

<http://www.tvdi.net>

10. OSGI Alliance.

<http://www.osgi.org>

11. Centro Nacional de Información y Comunicación Educativa (CNICE)

<http://www.cnice.mecd.es/>

12. Departamento de "e-learning" del CESGA

<http://www.cesga.es/ca/defaultC.html>

13. Educasites.net

<http://www.educasites.net>

14. Educaweb.com

<http://www.educaweb.com>

15. Eduforge.org

<http://eduforge.org>

16. Elearningpost

<http://www.elearningpost.com/elmisc/about.asp>

17. Elearnspace

<http://www.elearnspace.org/Articles/index.htm>

18. European Distance and E-Learning Network

<http://www.eden-online.org/eden.php?menuId=1>

19. European Training Village

<http://www.trainingvillage.gr/etv/default.asp>

20. RedIRIS

<http://rediris.es/>

8.2. Revistas

A continuación se listan algunas revistas sobre educación y eLearning, incluyendo su URL, factor de impacto y lugar en el ISI.

1. The Journal of the Learning Sciences

<http://www.editorialmanager.com/jls/>

EDUCATION & EDUCATIONAL RESEARCH (F.I. 2.792) (1/98)

2. Review of Educational Research (RER)

<http://www.jstor.org/journals/00346543.html>

EDUCATION & EDUCATIONAL RESEARCH (F.I. 1.76) (2/98)

3. Instructional Science (INSTR SCI)

<http://www.springerlink.com/content/102905/>

EDUCATION & EDUCATIONAL RESEARCH (F.I. 1.00) (13/98)

4. COMPUTERS & EDUCATION

http://www.elsevier.com/wps/find/journaldescription.cws_home/347/description

EDUCATION & EDUCATIONAL RESEARCH (F.I. 0.968) (16/98)

5. The British Journal of Educational Technology (BRIT J EDUC TECHNOL)

<http://www.blackwellpublishing.com/journal.asp?ref=0007-1013>

EDUCATION & EDUCATIONAL RESEARCH (F.I. 0.593) (33/98)

6. The Journal of Computer Assisted Learning (J COMPUT ASSIST LEAR)

<http://www.blackwellpublishing.com/journal.asp?ref=0266-4909>

EDUCATION & EDUCATIONAL RESEARCH (F.I. 0.556) (37/98)

También cabe destacar:

IEEE Transactions on Education.

<http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=13>

(página final)

Sevilla, Junio de 2007