UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE INFORMÁTICA

Tesis Doctoral en Ingeniería Informática

Extension of the IMS Learning Design Specification based on Adaptation and Integration of Units of Learning

Extensión de la especificación IMS Learning Design desde la Adaptación e Integración de Unidades de Aprendizaje

18 de enero de 2008
Leganés, Madrid, Spain

Autor: Daniel Burgos

Directores: Dra. Belén Ruiz Mezcua
Dr. Ángel García Crespo

Para David. Por tu fortaleza, tu determinación y tu apoyo incondicional

studio fons curiositatis est

curiositas fons agnitionis est

agnitio fons discis est

disco fons sapientiae est

sapientia fons temperationis est

temperatio fons vitae est

Acknowledgements / Agradecimientos

Esta investigación ha sido posible gracias a múltiples colaboraciones, trabajos conjuntos y apoyos incondicionales, algunos de los cuales continúan. Nunca me dejo de asombrar de la cantidad de gente con la que uno se relaciona y que influye en el trabajo propio, afortunadamente. Mis disculpas si alguien permanece en el tintero, algo completamente ajeno a mi intención. Mis agradecimientos son para:

- mis directores de tesis, Ángel García Crespo y especialmente Belén Ruiz Mezcua, con quien comencé este proyecto hace varios años

- el apoyo continuo del Tercer Ciclo de la Carlos III, personalizado en Julia Pasamar

- mi colega Colin Tattersall, con quien he realizado gran parte de esta investigación sobre IMS Learning Design en la Open University of The Netherlands y que tanto soporte profesional y personal me proporciona siempre

- mis actuales compañeros de trabajo Lydia Montandon, Daniel Fernández y José Luis Santos, de ATOS Origin/Research & Innovation, con los que estoy desarrollando algunos de los proyectos mencionados y utilizados en la tesis y que representan un buen ejemplo de trabajo colaborativo y complicidad profesional

- los coautores reales de varios artículos publicados relacionados con la tesis, algunos de los cuales pueden encontrarse en los anexos: Yongwu Miao, por sus interesantes conversaciones sobre IMS-LD a altas horas de la noche en Heerlen; Christof van Nimwegen, por su complicidad para trabajar con juegos educativos y planificadores en Action Script y su pasión compartida por la cocina; Pablo Moreno, con quien tuve el placer de trabajar y debatir sobre la manera de comunicar IMS-LD con el exterior y con quien compito por nuestra adicción al chocolate; Baltasar Fernández Manjón y José Luis Sierra, un gran equipo con quien da gusto compartir esfuerzos; Tim Sodhi, por su interés sobre interfaces visuales; Anders Berggren y Josep M. Fontana, con quienes comencé a investigar sobre la integración de IMS-LD y Moodle; y una línea especial para Gemma Corbalán Pérez, amiga incombustible, con quien comparto la pasión por el aprendizaje

- mi colega Jesús G. Boticario, coordinador científico del proyecto europeo EU4ALL, por su visión crítica y científica sobre aspectos punteros de aprendizaje y accesibilidad y por sus estimulantes conversaciones sobre temas ajenos al trabajo

- los profesores Carlos Delgado Kloos, Juanma Dodero y Paloma Martínez, que tan amablemente contribuyeron a la predefensa de esta tesis y que han proporcionado puntos de vista y consejos muy interesantes

- mis compañeros de fatigas en UNFOLD, germen conceptual de todo esto, Dai Griffiths y Josep Blat. Gracias al trabajo que ellos lideraron diligentemente IMS-LD se ha expandido notablemente en Europa

- los editores de las revistas científicas David Crookall (Simulation & Gaming, University of Nice-Sophia Antipolis), Patrick McAndrew (Journal of Interactive Media in Education, The Open University UK) y Robert Tennyson (Computers in Human Behavior, Stanford University), con quien he tenido el placer de trabajar como editor invitado en números especiales relacionados con esta investigación

- mis colegas Ambjörn Naeve (Royal Institute of Stockholm) y Eduardo Sánchez (Universidad de Santiago de Compostela), con quienes he trabajado en diversas ocasiones intentando aportar una visión innovadora sobre aprendizaje adaptativo y sobre IMS-LD

- los investigadores europeos Daniel Olmedilla (University of Hannover – L3S) y Carlos Velasco (Fraunhofer Institute - FIT), que tan amablente han accedido a revisar mi tesis. Ambos son compañeros en diversos proyectos y comparten el gusto por el trabajo preciso y bien hecho. Este agradecimiento se hace extensivo al doctor Luis Weruaga (Commission for Scientific Visualisation - Austrian Academy of Sciences), que también ha realizado una revisión concienzuda del texto

- mis amigos Gregorio Rodríguez y Marisol Ibarra (Cádiz), Marco van Hout (Maastricht), Laura Chavarría (Utrecht), Alberto Palmero (Sevilla), Martin and Reisi (Utrecht), por su contagiosa actitud positiva y sincero apoyo

- otros tantos compañeros con los que he colaborado o colaboro de alguna manera en tecnología educativa, quienes mayormente comparten la vocación por mejorar los procesos de aprendizaje y de enseñanza. He tenido la oportunidad de trabajar con todos ellos y de aprender en diversos frentes. En orden alfabético: Alejandro Rodríguez Ascaso (UNED), Alfredo Fernández Valmayor (Universidad Complutense de Madrid), Christian Martel (Université de Savoie), Elena del Campo (UNED), Enrique Gómez (Universidad Politécnica de Madrid), Ernie Giglione (LAMS), Griff Richards (Simon Fraser University), Hans Hummel (Open University of The Netherlands-OUNL), Hubert Vogten (OUNL), James Dalziel (LAMS), Laurence Vignolet (Université de Savoie), Luca Botturi (Università della Svizzera italiana), Luis Panizo (Universidad de León), Manuel Lama Penín (Universidad de Santiago de Compostela), Mar Saneiro (UNED), Marcus Specht (OUNL), Martin Dougiamas (Moodle), Milos Kravcik (OUNL), Paul De Bra (Technological Institute of Eindhoven – TUe), Rob Koper (OUNL).

Y, lo más importante, para el final. Gracias a Rocío por su comprensión infinita hacia mis necesidades de estudio y las tiranías de agenda. Admiro su habilidad para disfrutar de las cosas sencillas y para conseguir puntos de encuentro que disfrutar juntos.

Daniel Burgos

Barcelona, España

18 de enero de 2008

Abstract

IMS Learning Design (IMS-LD) is a current asset in eLearning and blended learning, due to several reasons:

a) It is a specification that points to standardization and modeling of learning processes, and not just content; at the same time, it is focused on the re-use of the information packages in several contexts;

b) It shows a deeper pedagogical expressiveness than other specifications, already delivered or in due process

c) It is integrated at different levels into well-known Learning Management Systems (LMSs)

d) There are a huge amount of European research projects and groups working with it, which aims at sustainability (in academia, at least)

Nevertheless, IMS-LD is roughly an initial outcome (be aware that we are still working with the same release, dated on 2003). Therefore, it can and must be improved in several aspects, i.e., pedagogical expressiveness and interoperability. In this thesis, we concentrate on Adaptive Learning (or Personalised Learning) and on the Integration of Units of Learning (UoLs). They both are core aspects which the specification is built upon. They also can improve it significantly. Adaptation makes personalised learning itineraries, adapted to every role, to every user involved in the process, and focus on several aspects, i.e., flow, content and interface. Integration fosters the re-use of IMS-LD information packages in different contexts and connects both-ways UoLs with other specifications, models and LMSs. In order to achive these goals we carry out a three-phase analysis. First, analysis of IMS-LD in several steps: foundations, information model, construction of UoLs. From Level A to Level C, we analyse and review the specification structure. We lean on a theoretical frameword, along with a practical approach, coming from the actual modeling of real UoLs which give an important report back. Out of this analysis we get a report on the general structure of IMS-LD.

Second, analysis and review of the integration of UoLs with several LMSs, models and specifications: we analyse three different types of integration: a) minimal integration, with a simple link between parts; b) embedded integration, with a marriage of both parts in a single information package; and d) full integration, sharing variables and states between parts. In this step, we also show different case studies and report our partial conclusions.

And third, analysis and review of how IMS-LD models adaptive learning: we define, classify and explain several types of adaptation and we approach them with the

specificacion. A key part of this step is the actual modeling of UoLs showing adaptive learning processes. We highlight pros and cons and stress drawbacks and weak points that could be improved in IMS-LD to support adaptation, but also general learning processes

Out of this three-step analysis carried out so far (namely general, integration, adaptation) we focus our review of the IMS-LD structure and information model on two blocks: Modeling and Architecture. Modeling is focused on process, components and programming resources of IMS-LD. Architecture is focused on the communication that IMS-LD establishes outside, both ways, and it deals with upper layers of the specification, beyong modeling issues. Modeling and Architecture issues need to be addressed in order to improve the pedagogical expressiveness and the integration of IMS-LD. Furthermore, we provide an orchestrated solution which meets these goals. We develop a structured and organized group of modifications and extensions of IMS-LD, which match the different reported problems issues. We suggest modifications, extensions and addition of different elements, aiming at the strength of the specification on adaptation and integration, along with general interest issues.

The main conclusion out of this research is that IMS-LD needs a re-structure and a modification of some elements. It also needs to incorporate new ones. Both actions (modification and extension) are the key to improve the pedagogical expressiveness and the integration with other specifications and eLearning systems. Both actions aim at two clear objectives in the definition of IMS-LD: the personalisation of learning processes, and a real interoperability. It is fair to highlight the welcome help of high-level visual authoring tools. They can support a smoother modeling process that could focus on pedagogical issues and not on technical ones, so that a broad target group made of teachers, learning designers, content creators and pedagogues could make use of the specification in a simpler way. However, this criticism is outside the specification, so outside the core of this thesis too.

This three-year research (2004-2007) has been carried out along with colleagues from The Open University of The Netherlands, The University of Bolton, Universitat Pompeu Fabra and from the Department of Research & Innovation of ATOS Origin. In addition, a few European projects, like UNFOLD, EU4ALL and ProLearn, have partially supported it.

Resumen

IMS Learning Design (IMS-LD) representa una corriente actual en aprendizaje online y blended que se caracteriza porque:

a) Es una especificación que pretende estandarizar procesos de aprendizaje, así como reutilizarlos en diversos contextos

b) Posee una expresividad pedagógica más elaborada que desarrollos anteriores o en proceso

c) Mantiene una relación cordial y prometedora con *Learning Management Systems* (LMSs), herramientas de autoría y de ejecución

d) Existe una amplia variedad de grupos de investigación y proyectos europeos trabajando sobre ella, lo que augura una sostenibilidad, al menos académica

Aun así, IMS Learning Design es un producto inicial (se encuentra en su primera versión, de 2003) y mejorable en diversos aspectos, como son la expresividad pedagógica y la interoperabilidad. En concreto, en esta tesis nos centramos en el aprendizaje adaptativo o personalizado y en la integración de Unidades de Aprendizaje, como dos de los pilares que definen la especificación, y que al mismo tiempo la potencian considerablemente. El primero (aprendizaje adaptativo) hace que se puedan abordar itinerarios individuales personalizados de estudio, tanto en flujo de aprendizaje como en contenido o interfaz; el segundo (integración) permite romper el aislamiento de los paquetes de información o cursos (Unidades de Aprendizaje, UoL) y establecer un diálogo con otros sistemas (LMSs), modelos y estándares, así como una reutilización de dichas UoLs en diversos contextos.

En esta tesis realizamos un estudio de la especificación desde la base, analizando su modelo de información y cómo se construyen Unidades de Aprendizaje. Desde el Nivel A al Nivel C analizamos y criticamos la estructura de la especificación basándonos en un estudio teórico y una investigación práctica fruto del modelado de Unidades de Aprendizaje reales y ejecutables que nos proporcionan una información muy útil de base, y que mayormente adjuntamos en los anexos, para no interferir en el flujo de lectura del cuerpo principal. A partir de este estudio, analizamos la integración de Unidades de Aprendizaje con otros sistemas y especificaciones, abarcando desde la integración mínima mediante un enlace directo hasta la compartición de variables y estados que permiten una comunicación en tiempo real de ambas partes. Exponemos aquí también las conclusiones de diversos casos de estudio basados en adaptación que se anexan al final de la tesis y que se vuelven un instrumento imprescindible para lograr una solución real y aplicable. Como segundo pilar de la tesis complementario a la integración de

Unidades de Aprendizaje, estudiamos el aprendizaje adaptativo: Los tipos, los avances y los enfoques y restricciones de modelado dentro de IMS-LD. Por último, y como complemento de la investigación teórica, a través de diversos casos prácticos estudiamos la manera en que IMS-LD modela la perzonalización del aprendizaje y hasta qué punto.

Este primer bloque de análisis (general, integración y aprendizaje adaptativo) nos permite realizar una crítica estructural de IMS-LD en dos grandes apartados: Modelado y Arquitectura. Modelado apunta cuestiones que necesitan mejora, modificación, extensión o incorporación de elementos de modelado dentro de IMS-LD, como son procesos, componentes y recursos de programación. Arquitectura engloba otras cuestiones centradas en la comunicación que realiza IMS-LD con el exterior y que apuntan directamente a capas estructurales de la especificación, más allá del modelado. Aunque se encuentra fuera del núcleo de esta tesis, también se ha realizado una revisión de aspectos relacionados con Herramientas de autoría, por ser este un aspecto que condiciona el alcance del modelado y la penetración de la especificación en los distintos públicos objetivo. Sobre Herramientas, no obstante, no realizamos ninguna propuesta de mejora.

La solución desarrollada, se centra en las diversas cuestiones sobre Modelado y Arquitectura encontradas en el análisis. Esta solución se compone de un conjunto de propuestas de estructuras, nuevas o ya existentes y modificadas, a través de las que se refuerza la capacidad expresiva de la especificación y la capacidad de interacción con un entorno de trabajo ajeno.

Esta investigación de tres años ha sido llevada a cabo entre 2004 y 2007, principalmente con colegas de The Open University of The Netherlands, The University of Bolton, Universitat Pompeu Fabra y del departamento Research & Innovation de ATOS Origin, y ha sido desarrollada parcialmente dentro de proyectos europeos como UNFOLD, EU4ALL y ProLearn. La conclusión principal que se extrae de esta investigación es que IMS-LD necesita una reestructuración y modificación de ciertos elementos, así como la incorporación de otros nuevos, para mejorar una expresividad pedagógica y una capacidad de integración con otros sistemas de aprendizaje y estándares eLearning, si se pretenden alcanzar dos de los objetivos principales establecidos de base en la definición de esta especificación: La personalización del proceso de aprendizaje y la interoperabilidad real. Aun así, es cierto que la implantación de la especificación se vería claramente mejorada si existieran unas herramientas de más alto nivel (preferiblemente con planteamiento visual) que permitieran un modelado sencillo por parte de los usuarios finales reales de este tipo de especificaciones, como son los profesores, los creadores de contenido y los pedagogos-didactas que diseñan la experienicia de aprendizaje. Este punto, no obstante, es ajeno a la especificación y afecta a la interpretación que de la misma realizan los grupos de investigación y compañías que desarrollan soluciones de autoría.

Nota: Por requerimientos oficiales del Ministerio de Educación, particulares de la Universidad Carlos III y específicos de este Programa de Doctorado, para la obtención del doctorado europeo se requiere que esta tesis tenga capítulos en inglés y en castellano, en una determinada proporción. Aparte del resumen y conclusiones en ambos idiomas: a) la introducción, el análisis general de la especificación, la definición de estructuras de modelado y arquitectura y la propuesta de solución se realizan en castellano, y b) el análisis específico de la especificación basado en el aprendizaje adaptativo y la integración de unidades de aprendizaje se realiza en inglés.

Table of contents/ Índice

Índice de ilustraciones

Índice de tablas

1. Introducción

1.1 . Objeto de estudio

IMS Learning Design, IMS-LD de ahora en adelante [1], es una especificación centrada en formación online (o *eLearning*), aunque también es utilizada en entornos mixtos (o *bLearning*) y que permite modelar programaciones curriculares o lecciones presenciales de forma que puedan ser seguidas individual o grupalmente, construyendo lo que se denomina Unidades de Aprendizaje (Units of Learning, UoL). También permite crear itinerarios de aprendizaje personalizados o adaptativos. IMS-LD puede representar una gran variedad de modelos pedagógicos y permite que el profesor o profesora adapte sus recursos y sus programaciones de aula a clases virtuales de una manera completamente flexible. Del mismo modo, IMS-LD permite que el estudiante consiga una experiencia completamente personalizada en función de su rendimiento o de su propia elección, adaptando los contenidos, los servicios proporcionados y el itinerario formativo. Lejos de mostrar actividades únicamente de manera secuencial o utilizar únicamente repositorios de objetos de aprendizaje, IMS-LD proporciona diversas características para generar aprendizaje adaptativo, dinámico y personalizado [2]. Mediante la descripción de diferentes roles, actividades, entornos, métodos, propiedades, condiciones y notificaciones, puede utilizarse para transformar las planificaciones de aula en Unidades de Aprendizaje formales (UoL). Actualmente existen diversas herramientas que arropan esta especificación, como CopperCore [3], el primer motor capaz de ejecutar UoLs creadas con IMS-LD, o los editores CopperAuthor [4] o Reload LD Editor [5], que ayudan y asisten en el proceso creativo de construir planes curriculares y gestionar los recursos. El profesor o diseñador de aprendizaje puede utilizar también ejemplos ya creados y adaptarlos a sus propios objetivos y estructura. Repositorios como Learning Networks for Learning Design [6] o Dspace [7] proporcionan algunos de estos paquetes de información listos para usar. En resumen, IMS-LD ayuda al profesor y al diseñador de aprendizaje a crear itinerarios pedagógicos online suficientemente flexibles para representar una amplia gama de objetivos, estilos y métodos pedagógicos.

No obstante, IMS-LD presenta ciertas características que limitan este grado de flexibilidad. Los sistemas de adaptación del aprendizaje, base de cualquier aprendizaje personalizado y esencia misma de la formación online, no son obvios y los recursos de programación son incompletos. Del mismo modo, la comunicación e interoperabilidad con otros sistemas o módulos externos o incluso recursos que cumplen el mismo esquema IMS-LD es un reto pendiente, ya que en las corrientes social y tecnológica actuales (código abierto u *open source*, código libre o *free source*, implementaciones multiplataforma...) los sistemas y formatos propietario aislados no son el único referente y no representan la tendencia mayoritaria actual, y menos en el ámbito educativo. Integración y adaptación de Unidades de Aprendizaje son dos aspectos que pueden y deben ser abordados dentro de la especificación y, por lo tanto, es la especificación la que puede dar respuesta mediante estructuras y propuestas concretas.

Adicionalmente, existen algunas características complementarias e importantes que afectan a IMS-LD aunque son responsabilidad del entorno formado por editores, visualizadores y motores de ejecución y no de la especificación en sí misma, como a) una gestión más eficaz de usuarios y roles, b) una ejecución interpretada y no compilada de los paquetes de información lo que conllevaría la moficación de contenidos y estructuras en tiempo de ejecución, c) unos editores gráficos de alto nivel, d) una gestión más transparente y bidireccional de la base de datos del motor de ejecución, e) una existencia de patrones y escenarios de Unidades de Aprendizaje pueden reforzar la utilización real de esta especificación, o f) un formato habitual de exportación de datos de contenido, método y roles/usuarios (TXT, CSV...).

Estos retos y necesidades no son exclusivos de IMS-LD, sino que se encuentran presentes en cualquier sistema, especificación o aplicación sobre aprendizaje online que trabaje con procesos y no únicamente con contenidos. Tanto editores (Cosmos [45], Dreamweaver Extension, MOT+ [8], Reload CP and Scorm Editors [5]), como motores de ejecución (CopperCore [3], netUniversité [9]), como sistemas gestores de cursos (WebCT-Blackboard, Claroline, Moodle [10]), como estándares *eLearning* (Scorm [11], IMS Simple Sequencing [12]), como sistemas integrados que combinen dos o más de las categorías anteriores (Lams [13] –autoría y LMS-, .LRN [14] –LMS y motor de ejecución-, Authorware –autoría y ejecución-), como la práctica educativa diaria reclaman la implementación de estas necesidades.

Nos centraremos principalmente en la integración y en la adaptación de Unidades de Aprendizaje (lo que supone abordar la interoperabilidad de contenidos y tecnología y la personalización del aprendizaje), base ambos del núcleo de la formación online mediante estándares que son a su vez responsabilidad directa de IMS-LD. Del mismo modo, constituyen estos aspectos dos de los principales retos que, a su vez, afectan e implican otras diversas necesidades colaterales ya mencionadas y que, sin embargo, implican otros recursos relacionados pero ajenos a la especificación en sí, como editores, motores de ejecución o práctica educativa. Fruto de esta crítica, propondremos las modificaciones y ampliaciones que sean necesarias para dotar a la especificación de una respuesta educativa eficaz mediante los recursos y estructuras tecnológicas adecuadas.

1.2 . Descripción del problema

IMS-LD es heredero de EML [29] y constituye el referente con más versatilidad en el mercado de especificaciones y lenguajes de modelado educacionales. Su capacidad para definir no únicamente contenido formativo y actividades asociadas, sino los procesos y flujos de aprendizaje y los roles implicados, hacen de él un producto polivalente. Aun así, IMS-LD presenta ciertas características que limitan su expresividad y su capacidad de adaptación en el contexto de los requerimientos actuales del aprendizaje online. Estas limitaciones se obtienen fruto de una fase de análisis práctico basado en el diseño, creación, ejecución y evaluación de Unidades de Aprendizaje, realizado desde 2004 con usuarios de diversas redes de aprendizaje, talleres y cursos sobre IMS-LD, y que puede encontrarse de manera extensa en los anexos (Sección 10). En concreto, se aprecian tres carencias principales:

a) Las limitaciones de las estructuras de modelado de propósito general y de su funcionalidad (Secciones 2 y 4). IMS-LD permite el modelado de Unidades de Aprendizaje centrándose en contenidos y procesos. Por definición, no suscribe ningún enfoque pedagógico concreto y aspira a modelar cualquiera. En la práctica, las estructuras proporcionadas en los tres niveles de la especificación (A, B y C) limitan la amplitud, profundidad y diversidad de dicho modelado. Por ejemplo, entre otras:

- Definición ambigua de la utilización del servicio de monitorización, que pueden ser modelados utilizando el esquema provisto por IMS-LD dentro de un entorno, pero también pueden ser utilizados mediante elementos globales ajenos a dichos entornos
- La monitorización de usuarios es únicamente individual, no existiendo ningún mecanismo de monitorización ni grupal ni de rol
- Ausencia de iteraciones con condición de entrada, condición de salida o número específico de repeticiones
- Únicamente se contempla el tiempo relativo de ejecución desde el momento de publicación, pero no el relativo a instanciación de la unidad de aprendizaje o ejecución de rol o usuario, lo que supone un uso limitado del tiempo, una ausencia de tiempo absoluto y una omisión de mecanismos de sincronización entre usuarios/roles basados en esta posible entrada
- La notificación de nivel C contempla únicamente el envío de un correo SMTP o el redireccionamiento del flujo de aprendizaje a una actividad. Asimismo, únicamente se puede llamar desde un punto de entrada. El uso y la funcionalidad son muy limitadas

b) Las estructuras y recursos de adaptación, base de cualquier aprendizaje personalizado y esencia misma de la formación online y mixta (blended learning) no se definen con IMS-LD de manera expresa, carecen de solución de modelado, y los recursos de programación relacionados son incompletos (Sección 4). Es decir, IMS-LD establece el aprendizaje personalizado como uno de sus objetivos fundamentales, pero no proporciona medios específicos que permitan modelarlo, sino que este modelado debe deducirse de los recursos generales existentes. Estas estructuras y procesos utilizan y combinan las estructuras de propósito general desarrolladas en el apartado previo y, adicionalmente, son creadas *ex profeso* para soportar procesos adaptativos específicos. Por ejemplo:

- Uso de condiciones simples en cascada para la definición de reglas de comportamiento y adaptación. Ausencia de condición múltiple con rangos
- Estructura de cálculo de aritmética simple mediante operandos por pares en cascada. Ausencia de una forma clara de realizar fórmulas de cálculo aritmético, imprescindible para la definición de reglas de comportamiento y evaluación adaptativa, por ejemplo
- No existen puntos de entrada o sincronización de flujo, estado o actividad
- No se puede realizar la asignación directa de una actividad o servicio a un usuario determinado
- No existe la posibilidad de formación dinámica de grupos dentro del mismo rol ni de grupos con miembros provenientes de distintos roles y que enriquezcan la adaptación individual con un flujo de comunicación bidireccional con un grupo de estudio
- No hay posibilidad de modificar el esqueleto, el método, la definición de roles o cualquier otro elemento estructural en la fase de ejecución o run-time

c) La integración, comunicación e interoperabilidad con otras especificaciones, sistemas de aprendizaje, módulos externos de sistemas de aprendizaje, o incluso recursos que cumplen el mismo esquema IMS-LD, es mínima o inexistente (Sección 3). En la corriente social y tecnológica actual (código abierto u open source, código libre o free source, implementaciones multiplataforma…) los sistemas y formatos propietarios aislados no son el único referente y no representan la tendencia mayoritaria actual. Menos aún en el ámbito educativo. Faltan estructuras de modelado y recursos técnicos estructurales asociados que den respuestas a esta carencia y posibiliten y faciliten una mejor integración e interoperabilidad de IMS-LD con otras especificaciones y con sistemas de aprendizaje. Por ejemplo:

- Carencia de tipos de propiedades que permitan el intercambio (importación, exportación) efectivo de datos con aplicaciones externas
- Ausencia de un sistema de grabación y recuperación de datos en y desde ficheros externos de ningún formato
- Ausencia de las conexiones con bases de datos externas o con módulos programados o modelados en otros lenguajes, notaciones o sistemas
- Ausencia de un sistema o capa de comunicación que permita el intercambio efectivo de información, como variables y valores de variables y estado, lo que permitirá interactuar desde IMS-LD en el módulo externo y viceversa

Todas estas estructuras y funcionalidades (de los epígrafes a, b y c en esta misma sección), entre otras que puedan surgir, serán definidas y desarrolladas convenientemente en el transcurso de la tesis, dando respuesta a las carencias y debilidades que encuentra en IMS-LD la fase de análisis de esta investigación (Sección 5).

Resaltar que la Integración y la Adaptación de Unidades de Aprendizaje son dos aspectos que pueden y deben ser abordados dentro de la especificación (tal y como se argumenta en las Secciones 2.2 y 2.7 de este documento) y, por lo tanto, es la especificación la que puede y debe dar respuesta mediante estructuras y propuestas concretas.

1.3. **Problemática complementaria**

Adicionalmente, existen algunas características complementarias e importantes que afectan a IMS-LD aunque principalmente son responsabilidad del entorno formado por editores, visualizadores y motores de ejecución que interpretan la especificación. En menor medida, también algunos dependen de IMS-LD en sí misma. Dichas características condicionan la adecuada respuesta de IMS-LD a necesidades y requerimientos concretos del proceso educativo. En el transcurso de la investigación también se estudiarán estas características y hasta qué medida y cómo pueden ser abordadas, con el objeto de proporcionar una respuesta más amplia y detallada a temas complementarios de la interoperabilidad y la adaptación, que los condicionan o dependen de ellos. Por ejemplo:

a) Una gestión más eficaz de usuarios y roles, donde se puedan realizar altas, bajas, agrupamientos de más de cinco miembros por rol, agrupamientos dinámicos de usuarios del mismo rol, agrupamientos de usuarios de distinto rol, asignación de usuarios a roles en tiempo de ejecución, etcétera

b) Una ejecución interpretada y no compilada de los paquetes de información lo que conllevaría la modificación de contenidos y estructuras en tiempo de ejecución.

Aunque un lenguaje compilado no debe ser sinónimo de contenido estático, en la actual interpretación del modelo de información de IMS-LD, así es

c) Unas herramientas de autoría de alto nivel. La edición de Unidades de Aprendizaje es realizada por los usuarios activos del proceso de aprendizaje, mayormente profesores/tutores, diseñadores de aprendizaje/pedagogos y alumnos en menor medida. Ninguno de estos grupos debería tener la necesidad de poseer conocimientos técnicos específicos fuera de su ámbito de conocimiento ya que esto limita la traducción de sus teorías y procesos de aprendizaje a Unidades de Aprendizaje concretas. La adaptación puede ser factible técnicamente pero difícilmente demostrable en la realidad cotidiana de un aula virtual si no puede ser generada por el usuario cotidiano no necesariamente técnico

d) Una gestión más transparente y bi-direccional de la base de datos del motor de ejecución que permita la compartición de propiedades y estados con otras aplicaciones externas

e) Una existencia de patrones y escenarios de Unidades de Aprendizaje pueden reforzar la utilización real de esta especificación dado el alto conocimiento técnico que hay que poseer actualmente para una edición eficaz

f) Un formato habitual de exportación de datos de contenido, método y roles/usuarios (TXT, CSV…). Aunque IMS-LD viene definido por la interoperabilidad, a día de hoy se produce una falta de integración absoluta con sistemas de aprendizaje que permite la interacción mutua de ambas definiciones. Si bien es cierto que existen alguna iniciativas prometedoras (Moodle, LAMS, MOT+, .LRN…) estas se basan en la exportación, importación y ejecución de Unidades de Aprendizaje y no en la interacción con sus elementos de manera bi-direccional. Por tanto, no existe ninguna manera de compartición de datos de contenido, método y roles

Estos retos o necesidades no son exclusivas de IMS-LD, sino que se encuentran presentes en cualquier sistema, especificación o aplicación sobre aprendizaje online que trabaje con procesos y no únicamente con contenidos.

1.4 . **Propuesta de solución**

Por tanto, básicamente la problemática a la que se pretende dar respuesta se centra en:

a) Estructuras general insuficientes o de diseño ambiguo de modelado

b) Carencia de estructuras específicas para aprendizaje adaptativo

c) Interoperabilidad e integración con otras especificaciones y sistemas de aprendizaje mínima o inexistente

Consecuentemente, IMS-LD no cumple, o lo hace de manera muy elemental o insuficiente, con varios objetivos importantes de su definición. En este trabajo de investigación se han propuesto y validado soluciones específicas a los problemas detectados. Las soluciones son, en correspondencia con la lista anterior (Sección 6):

a) Desarrollo de nuevas estructuras y elementos de modelado de propósito general y modificación de la definición de estructuras existentes. Estos elementos y estructuras serán utilizados como parte de otros desarrollados de manera específica para sustentar procesos de aprendizaje adaptativo. Constituyen, pues, un conjunto básico para ser utilizado directamente o como parte de un enfoque de personalización

b) Desarrollo de estructuras específicas para aprendizaje adaptativo. Ciertos procesos no pueden ser abordados con la única definición de estructuras generales y necesitan la definición adicional de estructuras y elementos específicos. De esta manera se logra conseguir el objetivo de adaptación de una manera, y se describe de una manera más directa

c) Modificación y desarrollo de estructuras que posibiliten y favorezcan una integración e interoperabilidad satisfactorias. La comunicación y compartición de información sobre usuarios, actividades y procesos con aplicaciones externas (por ejemplo, sistemas de aprendizaje) es algo no implementado de manera efectiva aún y garantiza la utilización de esta especificación en entornos reales de aprendizaje

Con todo ello se define cómo puede dicha especificación ser modificada o extendida para lograr una respuesta suficiente y optimizada que proporcione un mayor grado de identificación entre objetivos educativos, recursos tecnológicos y resultados de aprendizaje en la práctica educativa real. En definitiva, una mejora de la expresividad pedagógica.

Por lo tanto, diseñamos y modelamos las modificaciones y extensiones de las estructuras existentes así como el desarrollo de cuantas nuevas estructuras sean necesarias para dotar a la especificación de una respuesta educativa eficaz mediante los recursos y estructuras tecnológicas adecuadas.

En concreto, desarrollamos extensiones y modificaciones basadas en dos categorías:

a) Modelado: Que aborda el information model y aspectos específicos de código, con modificación de determinadas estructuras y modificación de sintaxis, de gramática y funcionalidad de otras. Por ejemplo: estructuras condicionales, global elements y acceso a propiedades, servicio de monitorización

b) Arquitectura: Que se centra en el aspecto funcional de la especificación, en concreto, la comunicación e integración de IMS-LD con otros sistemas y plataformas externas. Por ejemplo: Grabación y recuperación de datos, enlace con bases de datos de cualquier tipo, modificación dinámica del *learning design*

Se busca la máxima integración y la máxima funcionalidad con el mínimo impacto dentro de la estructura posible, es decir, respetar al máximo la especificación tal y como es con sus definiciones actuales y realizar los cambios e incorporaciones mínimos necesarios para completarla y extendarla acorde con nuestro análisis. No se busca, por tanto, la realización de un lenguaje de soporte adjunto a IMS-LD, sino de una modificación de la propia especificación.

Adicionalmente, realizamos un estudio sobre herramientas que describe la relación de IMS-LD como especificación con su implementación con visualizadores, editores y motores, su ejecución y su uso. Este apartado se encuentra fuera del alcance del núcleo de esta tesis (en tanto en cuanto no afecta a la especificación en sí, sino a la manera en que esta es interpretada por la tecnología y los desarrolladores del momento) pero muy relacionado. En la tesis, nos centramos pues en la modificación estructural de la especificación y no en su concreción en herramientas concretas. Aun así, dada la relevancia que para la especificación tienen los aspectos de autoría y divulgación, en su momento incorporamos tres estudios relacionados con editores visuales, con definición de plantillas y escenarios de aprendizaje, y con aprendizaje colaborativo, todos ellos en calidad de material complementario y estrechamente relacionado (Sección 9).

1.5 . **Objetivos**

La definición de objetivos se enmarca dentro del ámbito de esta tesis, es decir, la especificación IMS Learning Design centrada en aprendizaje adaptativo e interoperabilidad e integración de Unidades de Aprendizaje.

El objetivo fundamental se centra en mejorar la expresividad pedagógica de IMS-LD, refinando estructuras de modelado y arquitectura existentes y desarrollando estructuras complementarias que permitan una formalización más precisa, versátil y sencilla de procesos de aprendizaje adaptativo y de mecanismos y procesos de integración con sistemas de aprendizaje y otras especificaciones.

En concreto, fruto del análisis y crítica de la especificación IMS-LD desde el estudio práctico de la creación y uso de Unidades de Aprendizaje, los objetivos son:

- Obtener aspectos (elementos y estructuras) de modelado de propósito general y arquitectura de la especificación que estén definidos de manera insuficiente, ambigua o ineficaz y diseñar su modificación. De manera complementaria, desarrollar nuevos elementos y estructuras que complementen esta faceta y extiendan IMS-LD

- Del mismo modo, desarrollar elementos y estructuras de IMS-LD que sirvan para formalizar procesos y recursos centrados en aprendizaje adaptativo. Este objetivo supone una extensión (ampliación) a la especificación
- Desarrollar elementos y estructuras que posibiliten y faciliten procesos de integración de IMS-LD en aplicaciones externas, genéricamente, comprendiendo estas tanto sistemas de aprendizaje, como otras notaciones o especificaciones eLearning

Todos estos objetivos están justificados mediante un análisis exhaustivo del estado de la cuestión y del contexto de investigación, así como por el análisis y crítica de la especificación a través del desarrollo de ejemplos prácticos de unidades IMS-LD basados en escenarios de aprendizaje tipo.

1.6 . **Metodología**

En esta tesis pretendemos analizar y criticar la especificación IMS-LD desde un punto de vista educativo y técnico, basándonos en la investigación teórica y en la realización concreta de Unidades de Aprendizaje que implementen o intenten implementar situaciones y necesidades educativas con el objetivo de forzar, modificar, extender y complementar la especificación para mejorar el aprendizaje online asociado. Distinguimos varias fases y partes:

Fase 1. Estado del arte. Definición del contexto

Sección 2. Análisis de la situación actual de las especificaciones *eLearning* y justificación de la elección de IMS-LD como objeto de la tesis

Fase 2. Análisis y crítica. Definición del problema

Sección 3. Análisis de las necesidades educativas online y definición de la dependencia con IMS-LD. Descripción y análisis de IMS-LD mediante la implementación de Unidades de Aprendizaje ejecutables directamente sin ningún tipo de modificación

Sección 4. Justificación y descripción de integración y adaptación como base de la investigación. Descripción de escenarios de aprendizaje e implementación de soluciones concretas y Unidades de Aprendizaje con la definición actual de IMS-LD. Crítica a los límites actuales

Sección 5. Análisis y crítica de otros aspectos técnicos de IMS-LD con implicación en la práctica educativa real que complementen el núcleo de la tesis sobre adaptación e integración de Unidades de Aprendizaje

Fase 3. Modificación y extensión. Propuesta de solución y desarrollo

Sección 6. Propuesta y desarrollo de modificaciones y ampliaciones de elementos y estructuras de modelado y arquitectura de IMS-LD con el objeto de mejorar su versatilidad, simplificar su uso y aumentar su capacidad y precisión expresivas en situaciones de aprendizaje adaptativo e integración de Unidades de Aprendizaje

Fase 4. Evaluación de la solución

Sección 7. Evaluación de la solución propuesta mediante los mecanismos descritos en la siguiente sección

Toda la Fase 2, "Análisis y crítica. Definición del problema", surge del modelado de Unidades de Aprendizaje llevado a cabo de manera independiente por el autor como parte de esta tesis. Al mismo tiempo, existen Unidades de Aprendizaje que el autor modela o co-desarrolla como parte de talleres sobre IMS-LD y proyectos de investigación y que responden a necesidades de problemas académicos y docentes modelados con la especificación (por ejemplo, la personalización del flujo de aprendizaje como apoyo a un aprendizaje accesible, que se desarrolla en el proyecto EU4ALL y que se detalla en la Sección 1.8.6, o la resolución técnica del nivel C de la especificación dentro del proyecto UNFOLD, descrito en la misma sección).

Estos casos de estudio que diseñan, desarrollan y analizan Unidades de Aprendizaje son fundamentales para estudiar en profundidad el planteamiento teórico de IMS-LD y cómo se concreta en soluciones específicas de modelado. Excepto el primero de muestra, todos los demás se encuentran detallados en los anexos, Sección 10. A través de los casos de estudio obtenemos un informe minucioso sobre qué estructuras y elementos faltan o pueden ser modificados para mejorar la expresividad pedagógica de la especificación que, en suma, constituye el fin último de esta tesis y que se refleja en la Sección 6.

En estos casos de estudio el proceso de modelado que se realiza suele incluir un trabajo conceptual previo a la programación en sí. Como consecuencia, se producen una serie de documentos u hojas de trabajo que ayudan en la comprensión del problema, en la definición de la solución y en el modelado posterior con la especificación. Generalmente, este *storyboard* se compone de cuatro tipos de hojas de trabajo y diagramas:

EU4ALL. SP5	
WP	WP5.1, WP5.2
Scenario title	Human resources live
Scenario ID	SC-003
Author/s	Daniel Burgos /Daniel Fernández
Date	13 July 2007
Version	1.1
Status	Draft

General definition

Focus	Y
Interface	
Learning flow	X
Content	
Interac. problem solving support	X
Adaptive evaluation	X
Changes on-the-fly	X
--	

Roles	n
Learner	n
Set of rules	1
Tutor	1

Learning category	Y
Remembering	
Understanding	
Applying	
Analysing	
Evaluating	X
Creating	X

General description and link to related WP/s and SP5
A human resources woman who has to create a quiz for a course and has to adapt it as long as the course takes place (in the runtime). This system allows for a better adaptation to the users' needs along the learning flow. Since there is almost no distinction between design time and run time, learning goals, objectives and content can be easily adapted on the run.

Learning flow . Step by step
1. Setting-up of the unit of learning with full personalization of questions, answers, right answers, ranges, points earned, messages of feedback and welcome, title, that can be adapted and re-published in runtime 2. Questions and related properties are local (loc-property) and keep the same value for all the users in the same run, but personal answers and calculations are private and linked to every participant (locpers-property) 3. Five questions with three possible answers. If the answer is right earns the amount of points defined in the set-up 4. Only when you have answered the five questions you can go ahead to see the results 5. There are 2 roles, teacher and participant, and the learning flow swaps between them: a. First, the teacher sets-up the questionnaire and the participant waits for the opening of the course. The teacher can have a preview of the questionnaire before publishing b. Second, the teacher publishes the quiz and the participant answers the questions. The teacher monitors his/her progress c. Third, the participant finishes the quiz and receives two inputs: an adaptive feedback and

None

General completed when
The tutor consider that the learners have successfully completed the questionnaires

Figure 1. Unidad de Aprendizaje: Narrativa de un escenario de aprendizaje

En la primera hoja de trabajo (Figure 1) se define la narrativa del escenario de aprendizaje que debe modelarse. Aquí se especifican las necesidades y el contexto, así como los roles involucrados, los objetivos y los requerimientos. También se aportan las categorías de aprendizaje y cualquier otro concepto teórico que ayude en la interiorización del problema.

EU4ALL. SP5	
WP	WP5.1, WP5.2
Scenario title	Human resources live
Scenario ID	SC-003
General description	A human resources woman who has to create a quiz for a course and has to adapt it as long as the course takes place (in the runtime). This system allows for a better adaptation to the users' needs along the learning
General objectives	...
General pre-requisites	None
General completed when	The tutor consider that the learners have successfully completed the questionnaires.
General feedback	Adaptive feedback and new actitivies.

Learning flow. Step by step
1. Setting-up of the unit of learning with full personalization of questions, answers, right answers, rang earned, messages of feedback and welcome, title, that can be adapted and re-published in runtime 2. Questions and related properties are local (loc-property) and keep the same value for all the users i run, but personal answers and calculations are private and linked to every participant (locpers-propert 3. Five questions with three possible answers. If the answer is right earns the amount of points define up 4. Only when you have answered the five questions you can go ahead to see the results 5. There are 2 roles, teacher and participant, and the learning flow swaps between them: a. First, the teacher sets-up the questionnaire and the participant waits for the opening of the course. can have a preview of the questionnaire before publishing b. Second, the teacher publishes the quiz and the participant answers the questions. The teacher mor progress

List of milestones

ID	Title	Type	Roles	Description	Objectives	Pre-requisites	Completed when	Feedback	Services	Content
ML-01	UoL set-up.	Activity	Tutor	Setting-up of the questions, answers, personalization, points earned and so on (described in the steps 1-4).	To create a custom learning environment.					link
ML-02	Open the course and publish questionnaires.	Activity	Tutor	The tutor opens the quiz and makes it available for the students.	To start the learning experience.					
ML-03	Fill in questionnaire	Activity	Learner	Learners fill in the questionnaire	To check what learners know			Positive formative feedback		
ML-04	Questionnaire finished?	Condition	Set of rules	While the learners don't finish the questionnaire.	To stop the flow until the questions have been finished.		Learners finish all questions.	Adaptive feedback and a new activity.	Monitoring, forum, notifications.	
ML-05	Assessment and monitoring	Synchronize	Tutor, Set of rules	The tutor and the set of rules evaluate the learners while they send their answers.	To evaluate the learner's performance					
ML-06	Results for the learner	Activity	Tutor	Adaptive feedback	To provide adaptive evaluation	The previous milestone is finished.			Forum, notifications...	
ML-07	Next activity	Activity	Learner	A new activity.	To go on the learning itinerary					

Figure 2. Unidad de Aprendizaje: Descripción de milestones

En la segunda hoja de trabajo (Figure 2) se migra la narrativa anterior a una definición de pasos o *milestones* que identifican los momentos, actividades, elementos y procesos clave del escenario de aprendizaje. Con este guión de bloques se establece un paso intermedio entre la narrativa y el diagrama posterior.

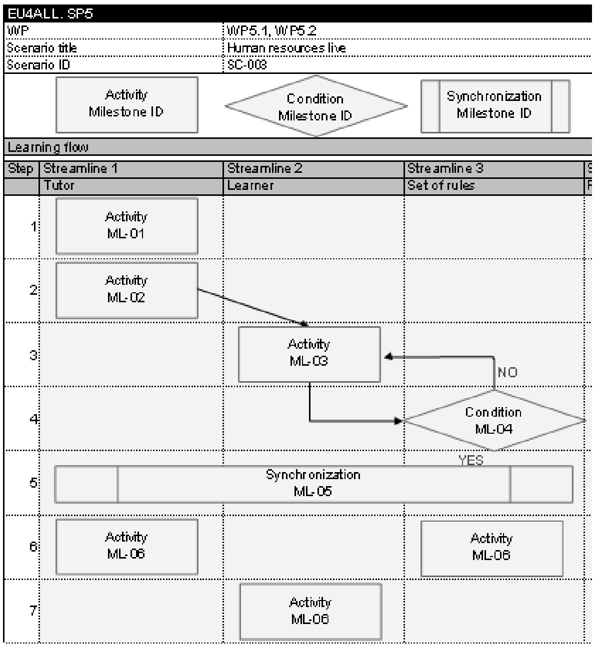


Figure 3. Unidad de Aprendizaje: Diagrama de flujo

En la tercera hoja de trabajo (Figure 3) se expresa el escenario, con sus actividades y procesos, en un flujo de aprendizaje que incluye condiciones y valores asociados. Gráficamente, enlaza los *milestones* en un hilo o itinerario de aprendizaje.

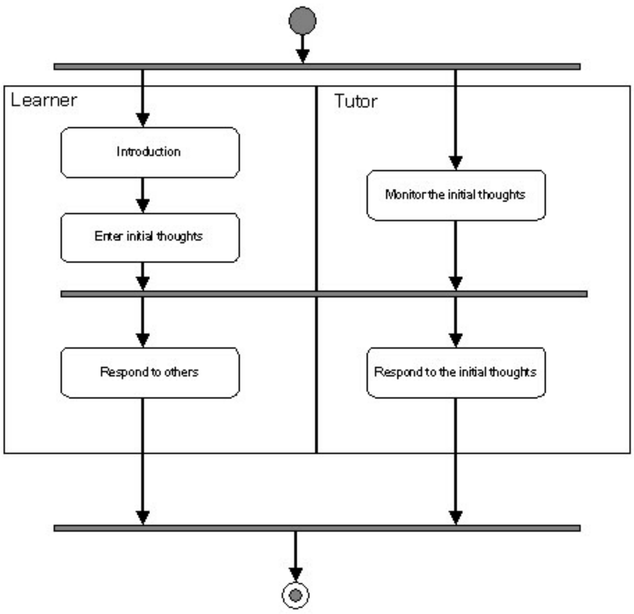


Figure 4. Unidad de Aprendizaje: Diagrama UML

Las tres hojas de trabajo anteriores (así como sus versiones previas) han sido determinantes a la hora de hacer que el público objetivo pudiera expresarse de una manera familiar pero precisa. Aun así, si bien para los perfiles más técnicos toda la contextualización y concreción de la narrativa teórica ha sido fundamental para no abordar directamente la problemática técnica y así poder encuadrar y categorizar problemas y soluciones posibles, existe la tendencia a utilizar notaciones más próximas a diagramas de flujo tradicionales. UML es el más habitual. Es por ello, que también se ha utilizado como cuarta hoja de trabajo (Figure 4), de manera opcional y complementaria.

1.7 . **Evaluación**

La evaluación de la tesis doctoral y de la solución desarrollada abarca dos grupos objetivo, tal y como se detallará en la Sección 7:

a) Parte consultiva con expertos en diseño EML. Se contacta con un selecto grupo de expertos en lenguajes de modelado y especificaciones eLearning y se recopila su opinión sobre la tesis, la solución de modelado y arquitectura y las unidades y escenarios de aprendizaje desarrollados en la parte empírica. Fundamentalmente, si estas extensiones con los objetivos propuestos y si las soluciones mejoran la expresividad de IMS-LD en adaptación e interoperabilidad

b) Parte consultiva con diseñadores de aprendizaje y tutores online en activo. Se contacta un amplio y variado grupo de usuarios de sistemas de aprendizaje con conocimiento de IMS-LD y se recopila su opinión y crítica sobre los escenarios de aprendizaje modelados en la parte empírica con las extensiones y modificaciones desarrolladas en la tesis. Al igual que con los expertos en diseño EML, se pretende valorar si se cumplen los objetivos y si las soluciones de modificación y extensión mejoran la adaptación e interoperabilidad que IMS-LD puede modelar; esta vez, desde el punto de vista práctico de los usuarios cotidianos de sistemas y procesos online en entornos reales de aprendizaje

1.8 . **Contexto de investigación**

1.8.1. **Estándar *versus* especificación**

Un estándar es una tecnología, formato o método, reconocido, nacional o internacionalmente, documentado en detalle y ratificado por una autoridad respetada de su campo, como ISO (International Standards Organisation), BSI (British Standards Institute), CEN (Centre Europeande Normalisation) o IEEE. Por el contrario, una especificación es el paso previo, creado por alguna compañía u organismo, que no ha sido ratificado todavía por ninguna autoridad, y que suele utilizarse de manera provisional pero suficientemente respaldada [15; 16].

Mientras que la especificación existe como tal es plenamente operativa y se produce mucha actividad corporativa e investigadora para identificar necesidades y carencias, para realizar re-revisiones y para depurarla lo más posible hasta que se obtiene el estándar [17] (Figure 5).

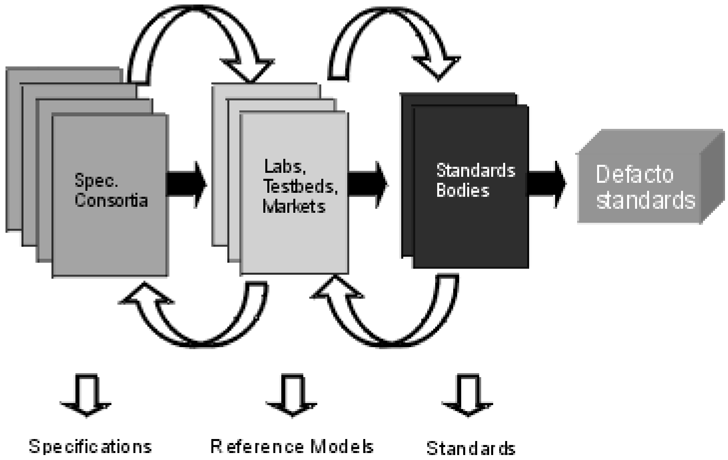


Figure 5. Proceso de generación de un estándar

Así pues, el proceso completo consiste en detectar una necesidad en un entorno profesional, crear una especificación para abordarla y, finalmente y si es necesario, estandarizarla. Las especificaciones actuales sobre *eLearning* funcionan como estándares de facto, es decir, no son todavía un estándar (de hecho, no existe ningún estándar sobre *eLearning* aún) pero las comunidades científica, académica y empresarial las consideran como tal y trabajan con ellas como si lo fueran.

Además de IMS Learning Design, existen otras especificaciones en torno al eLearning dentro y fuera del consorcio IMS. Citamos aquí las más importantes.

1.8.2. **Especificaciones IMS**

IMS desarrolla y mantiene más de veinte especificaciones distintas centradas en eLearning: Common Cartridge, Enterprise, Vocabulary, etc. Las más relacionadas con el diseño instructivo que también aborda IMS Learning Design son las siguientes [15; 16; 18].

*1.8.2.a. **IMS Simple Sequencing***

Esta especificación es la responsable de la definición de secuencias de las actividades de enseñanza/aprendizaje. En ella se definen las reglas que describen qué secuencia instructiva debe seguirse de acuerdo con las acciones realizadas por el usuario (estudiante).

La *Simple Sequencing Specification* [12] facilita los medios necesarios para representar la información utilizada para definir diversas posibilidades de secuencias de actividades educativas, a través de la descripción de diferentes rutas de navegación por una colección de materiales didácticos. Además, define el método para representar el comportamiento de un objeto de aprendizaje para que cualquier tecnología educativa compatible sea capaz de ejecutar las secuencias de actividades educativas de forma consistente.

1.8.2.b. ***IMS Content Packaging***

Normalmente, el contenido educativo necesita ser representado de manera automática de alguna manera informática, de forma que se permita la agregación, distribución, gestión y despliegue eficiente del mismo.

Por una parte, los autores de materiales educativos necesitan herramientas y tecnologías que los asistan en la creación de contenidos. Por otra parte, los fabricantes de sistemas de gestión de aprendizaje, los fabricantes de plataformas informáticas y los proveedores de servicios de aprendizaje quieren crear herramientas que garanticen una distribución y gestión eficiente de los materiales educativos creados por los autores. Los alumnos también necesitan herramientas de despliegue y de entrega que sean eficaces y de calidad.

Para satisfacer estas necesidades es importante que los contenidos estén empaquetados en una estructura y un formato de fichero conocidos, con una buena documentación de soporte. *IMS Content Packaging* satisface estas necesidades. *IMS Content Packaging* [19] se ocupa de la descripción, estructura y localización de materiales de aprendizaje en línea y de la definición de algunos tipos de contenido particulares. Permite que el autor encapsule todos los recursos requeridos, los sitúe en una estructura y añada los metadatos. De esta forma el usuario puede describir y empaquetar materiales de aprendizaje, tal que cursos individuales o colecciones de cursos, en paquetes distribuibles e interoperables.

La estructura *Content Packaging* puede integrar diseños de aprendizaje, tal como se muestra en la siguiente figura (Figure 6):

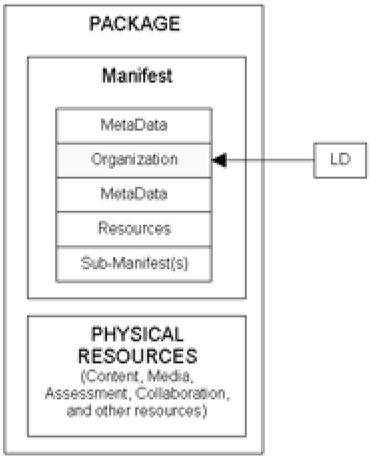


Figure 6. Paquete de información en IMS-CP (Fuente: IMS, 2001)

De esta forma, *Content Packaging* es la estructura que integra todos los elementos. Un *Content Package* agrupa LD (*Learning Design*), SS (*Simple Sequencing*), Metadata y QTI (*Question and Test Interoperability*). La Final Version 1.1.2 de la *IMS Content Packaging* Specification fue hecha pública en agosto de 2001. Actualmente la versión 1.4 se encuentra en fase de revisión para su publicación a lo largo de 2007/2008.

1.8.2.c. ***IMS Learner Information Package***

El *IMS Learner Information Package* [20] es una especificación de un método estándar para recopilar información sobre los estudiantes.

Fue diseñado con la finalidad de permitir que la información recogida sobre los estudiantes y su rendimiento pudiera ser transferida entre distintas instituciones y aplicaciones de software.

Por ejemplo, puede obtenerse un registro detallado del rendimiento y de los objetivos conseguidos por el estudiante. Esta manera de obtener información sobre el rendimiento de los estudiantes podría incluso substituir los certificados.

Podemos también guardar las preferencias del estudiante para así poder facilitar el trabajo de los estudiantes con discapacidades.

Toda la información sobre los estudiantes se guarda en un fichero XML. Este fichero utiliza etiquetas explicativas sobre el significado de la información contenida en cada registro.

1.8.2.d. ***IMS Question and Test Interoperability***

Question and Test Interoperability [21] es una especificación para conseguir una manera estándar de compartir avaluación y datos. Es una especificación XML interoperable para describir preguntas y tests. El objetivo del QTI es facilitar que la compartición de información de tipo preguntas, tests y resultados sea más sencilla. Con el fin de permitir a los usuarios importar y exportar preguntas, tests y resultados entre diferentes aplicaciones, IMS-QTI tiene que ser claro y conciso de esta manera se evitan ambigüedades. Esta especificación fue construida de forma que es posible soportar preguntas y tests tanto simples como complejos, y además ofrece las estructuras de datos necesarias para asegurar la *interoperabilidad*.

De esta manera la información sobre preguntas y sobre el estudiante o sus resultados se puede compartir entre diferentes sistemas de *learning management* y diferentes paquetes software.

Los autores de las evaluaciones pueden crear sus propias preguntas o incluir preguntas diseñadas por otros usuarios de IMS-QTI, haciendo más fácil crear repositorios de preguntas reutilizables en diferentes sistemas.

1.8.2.e. ***Otras especificaciones IMS sobre e-learning*** *[22]*

Publicación	**Especificación**
2004-August-23	IMS AccessForAll Meta-Data
2003-January-30	IMS Digital Repositories Specification
2004-August-24	IMS Enterprise Services
2002-July-16	IMS Enterprise Specification
2005-July-5	IMS ePortfolio
2005-March-1	IMS General Web Services
2000-May-05	IMS Learning Resource Meta-data Specification
2001-October-01	IMS Meta-data Specification
2000-June-05	IMS Question and Test Interoperability Specification
2004-August-30	IMS Resource List Interoperability

2002-October-25	IMS Reusable Definition of Competency or Educational Objective
2004-July-30	IMS Shareable State Persistence
2004-March-22	IMS Vocabulary Definition Exchange

1.8.3. Otras especificaciones eLearning fuera de IMS

1.8.3.a. ***SCORM***

El *Sharable Object Reference Model* o modelo de referencia de objetos de contenido compartibles [11] es parte de la estrategia de la iniciativa ADL (*Advanced Distributed Learning* o Aprendizaje Distribuido Avanzado).

SCORM define una manera de representar objetos de contenidos compartibles (SCOs o *Sharable Content Objects*) como estructuras en XML. Este marco permite que los usuarios definan y accedan a información sobre objetos de aprendizaje de tal forma que estos puedan ser fácilmente compartidos entre diferentes sistemas de gestión de aprendizaje. SCOs son Unidades de Aprendizaje individuales que pueden combinarse para crear un curso. Los SCOs deben ser:

- *Duraderos*: Recursos electrónicos que no necesitan ser actualizados o modificados si cambia la tecnología
- *Interoperables*: Recursos que pueden ser utilizados correctamente por diferentes VLEs (*Virtual Learning Environments* o entornos virtuales de aprendizaje)
- *De fácil acceso*: Recursos que pueden ser encontrados cuando se necesitan. Los SCOs tienen vinculada una descripción de su contenido, esta descripción es útil para facilitar su búsqueda
- *Reutilizables*: son desarrollados una única vez y utilizados en muchos cursos

En los SCOs el material de aprendizaje y los meta-datos son empaquetados y pueden ser importados y exportados entre diferentes VLEs.

Los documentos *SCORM* son técnicos y especifican las funcionalidades que los sistemas deben incorporar para ser conformes. No indican cómo crear buen material *SCORM* o aprendizaje electrónico efectivo.

SCORM es útil, pero tiene limitaciones. Sólo es aplicable a la instrucción multimedia, individual y sin ayuda que la formación asistida por ordenador ha realizado tradicionalmente. La formación guiada por un instructor no está dentro de los objetivos de *SCORM*.

SCORM es un marco para la Web y el aprendizaje apoyado por ordenador que es definido por directrices, especificaciones y estándares. *SCORM* no es en sí mismo una especificación o un estándar. Es un modelo que referencia un conjunto de especificaciones técnicas publicadas, estándares y guías. Esta colección consta de los siguientes elementos: modelo de agregación de contenidos (CAM o *Content Aggregation*), entorno de ejecución (RTE o *Run-Time Environment*) y secuencia y navegación (SN o *Sequence and Navigation*). El modelo describe la creación, despliegue y comportamiento de los SCOs cuando son ejecutados en un sistema de gestión de aprendizaje basado en web.

En primer lugar, *SCORM* define la manera en que deben crearse los SCOs. La creación comienza con los *assets* (elementos de contenido de más bajo nivel): Imágenes, textos, sonido, o cualquier objeto o elemento que pueda ser interpretado por un navegador. Estos *assets* son ensamblados en SCOs. Cada SCO tiene un documento con los meta-dados que definen los *assets*. Una vez que los SCOs han sido creados, *SCORM* define cómo tienen que comportarse en un VLE.

La agregación de contenidos es el proceso de creación, descripción y empaquetado de los SCOs en la estructura de un curso.

El comportamiento en tiempo de ejecución es el proceso de lanzamiento de un SCO en un VLE, y el seguimiento de las actividades de los alumnos con los SCOs.

Un fichero de manifiesto (*manifest*) se encarga del empaquetamiento del SCOs dentro de la estructura del curso. *SCORM* define que ningún SCO puede tener enlaces de navegación a otros SCOs, la estructura de navegación al completo debe ser definida en una tabla de contenido en el fichero de manifiesto.

En la Sección 3.2, utilizamos SCORM como especificación de referencia para la integración de paquetes de información con IMS-LD.

1.8.3.b. ***LOM***

El estándar *IEEE LTSC Learning Object Metadata* [23] especifica la sintaxis y la semántica para objetos de aprendizaje, siendo este una entidad digital o analógica, que puede ser utilizada, reutilizada o referenciada durante el aprendizaje asistido por tecnología. Ejemplos de este tipo de aprendizaje son los sitemas de formación basados en ordenador (CBT), los sistemas de aprendizaje a distancia y los entornos de aprendizaje colaborativo. Por su parte, objetos de aprendizaje pueden ser contenidos multimedia, contenidos instructivos, objetivos educativos, software instructivo, herramientas informáticas educativas y personas, organizaciones o eventos referidos en el transcurso del proceso de aprendizaje asistido por tecnología. LOM se centra en proporcionar el conjunto mínimo de atributos necesarios para que estos objetos de aprendizaje sean gestionados, localizados y evaluados de manera apropiada.

Algunos de los atributos más representativos de este estándar sobre metadatos pueden ser tipo de objeto, autor, propietario, condiciones de distribución, formato, estilo pedagógico, nivel de conocimientos y pre-requisitos. Cada conjunto de metadatos es una entidad en sí misma y un mismo objeto puede estar sujeto a varios conjuntos al mismo tiempo. Por ejemplo, uno describiendo la documentación, otro el estilo pedagógico y otro la parte administrativa. El objetivo último de utilización de LOM es integrarse con otros estándares ya definidos que complementen la definición y utilización precisas de cada objeto de aprendizaje.

1.8.3.c. ***Dublin Core***

La iniciativa *Dublin Core Metadata* [24] está vinculada al desarrollo de especificaciones sobre metadatos *online* que pueden ser utilizados por una amplia variedad de propuestas y modelos de negocios. Las actividades de DCMI incluyen el trabajo de grupos especifícos, conferencias globales, talleres, el desarrollo e implementación de estándares, así como iniciativas educacionales para la promoción del uso de los estándares de metadatos. El primer taller del *Dublin Core* se efectuó en Dublín en 1995. En sus inicios el *Dublin Core* fué desarrollado en inglés pero en la actualidad ya existen versiones en otros idiomas.

Los metadatos del *Dublin Core* son empleados como suplementos a los métodos existentes para la búsqueda e indexado de recursos web, puesto que estos metadatos ofrecen un vocabulario semántico que describe las propiedades fundamentales de tales recursos.

Los metadatos que describen un determinado recurso pueden estar incluidos en un fichero independiente (como el catálogo de la biblioteca) o dentro del propio recurso, como es el caso de los documentos HTML; en forma de pares nombre-valor almacenados como meta tags (etiquetas meta) en el encabezado del documento.

1.8.4. **Lenguajes de modelado educativo**

Los lenguajes que a continuación enumeramos no son especificaciones ni estándares sino esfuerzos de grupos de investigación por aportar un modelado conceptual a los procesos de aprendizaje. Es por ello, que figuran en una categoría distinta a IMS-LD [25]. Algunos de ellos realizan aportaciones muy interesantes al modelado instruccional aunque apenas relacionadas con el objeto de esta tesis. En cualquier caso, proporcionan un soporte teórico complementario. Todos ellos cumplen con la definición del modelo básico del CNN/ISSS WSLT de un lenguaje de modelado educativo (Rawlings et al., 2002), en el que se establecen las entidades y relaciones entre entidades involucradas en el proceso de aprendizaje: persona, propiedad, dossier, rol, actividad de aprendizaje, actividad de soporte, estructura de actividad, objetivo, prerrequisito, producto, recurso y entorno. Estos lenguajes suelen tener dos restricciones comunes importantes: 1) se orientan demasiado al contenido y prescinden de procesos de aprendizaje aparte del lineal o incluso de cualquier tipo de proceso (salvo en algún caso, como en MOT). Esto supone que procesos de aprendizaje más complicados no son posibles, como el aprendizaje adaptativo; y 2) los modelos definidos son propietarios y no contemplan la interoperabilidad con otros lenguajes o con especificaciones.

En orden alfabético:

1. Active Document [26], describe y diseña actividades educativas colaborativas en sistemas distribuidos. Se centra fundamentalmente en ciencias experimentales dividiendo las tareas en actividades definidas en conceptos de laboratorio, como curso, práctica de laboratorio o experimento

2. CDF [27], utiliza el Ariadna Course Description Format para la descripción de cursos que se componen de un conjunto de documentos XML junto con un generador de curso LMS. Enfatiza el contenido y su agregación aunque también puede abordar el proceso pedagógico en función de un modelo determinado. El contenido se limita a formato texto

3. CPM [28], desarrollado en la Université de Pau francesa, el Cooperative Problem based learning Metamodel se presenta como un perfil de UML directamente modelable en XML. Se divide en cuatro paquetes: básico, pedagógico, estructural y social. Su unidad básica es la actividad y utiliza facilidades de agregación e interacción para desarrollar problemas colaborativos

4. EML [29], desarrollada por la Open University de los Países Bajos, bajo la dirección de Rob Koper. Es el fundamento de IMS-LD y la que da nombre a la categoría de Educational Modelling Language, a la que pertenecen todos los lenguajes listados en esta sección. Centra su diseño en la actividad de estudio realizadas por los miembros de los roles en un entorno determinado

5. E2ML [30], desarrollado en Università Della Svizzera Italiana, el Educational Environment Modeling Language se centra en el desarrollo de patrones como definición de soluciones reutilizables que puedan ayudar a los diseñadores instruccionales. Existen cuatro tipos de documentos: Objetivos, Diagramas de Acción, Lista de Recursos y Diagramas de Conjunto

6. LDL [31], de la Universidad de Savoie, se centra en el diseño de actividades educativas, que son consideradas un conjunto de intercambios que se producen entre participantes en un contexto determinado. Estos contextos se definen en torno a contenidos, objetivos y reglas
7. LMML [32], se basa en un meta-modelo para ser utilizado en diferentes dominios de aplicación. Se centra en una estructura conceptual, modular y jerárquica del contenido y puede adaptarse a distintos alumnos y situaciones de aprendizaje
8. MISA, MOT y MOT+. MISA (Méthode d'Ingénieie d'un Système d'Apprentissage) es una metodología de diseño de unidades didácticas en la que se propone un meta-modelo propio [8]. Por su parte MOT (Modélisation par Objets Typés) proporciona una representación gráfica de los elementos de dicho meta-modelo (2003) que se compone, a su vez, de cuatro modelos: conocimiento, didáctico, materiales educativos y entrega. MOT+ pretende desarrollar gráficamente el modelo de IMS Learning Design [33].
9. PALO [34], fue desarrollado en la UNED y se centra en los contenidos, clasificados como objetos con relaciones semánticas. Las actividades se ocupan principalmente de la definición de la descripción de ejercicios y trabajos para los alumnos
10. Targeteam [35], se basa en la reutilziación y generación dirigida de materiales de aprendizaje. La unidad de estudio no es la actividad sino la "cuestión", aunque no aborda el proceso de aprendizaje. Permite la utilización de material en distintas situaciones y dominios
11. TML/Netquest [36], lenguaje diseñado para separar el contenido semántico de la compasión de una pregunta. Los archivos tienen formato de texto y pueden ser importados desde bases de datos
12. Xedu [37], modela unidades didácticas y diseña el sistema software para representarlas. Para ello, define un modelo de información y un modelo de comportamiento. Se centra en el proceso computacional y no así tanto en el lenguaje de modelado. Se centra pues en la representación XML textual

1.8.5. Estado actual de las herramientas IMS-LD

La descripción del estado actual de las herramientas IMS-LD proporciona una visión general sobre cuántas, cuáles y hasta qué nivel existen [38-40]. (Aunque estos documentos están fechados en 2005 reflejan muy bien el estado actual, ya que no ha habido ningún progreso reseñable desde entonces). El análisis principal muestra que el grado de desarrollo es moderadamente amplio pero desigual y, sobre todo, no homogéneo ni suficientemente probado. Aunque crece exponencialmente no existe una normalización, más que deseable, ni una experiencia real que aporten resultados significativos que permitan mejorar las herramientas y, si procede, modificar o extender la especificación convenientemente. De todas las herramientas únicamente dos o tres han sido utilizadas en experimentos controlados o como versiones beta en cursos de prueba sin que ninguna haya sido realmente utilizada en explotación real a día de hoy.

Desde que en febrero de 2003 se aprobó IMS-LD se han ido desarrollando un cierto número de herramientas: Editores, reproductores y motores de ejecución.

Una aplicación acorde con la especificación Diseño de Aprendizaje IMS es necesariamente muy compleja, y la construcción de un reproductor para este tipo de aplicaciones implica un esfuerzo de desarrollo muy importante. Coppercore [3], es un reproductor de Código Abierto capaz de ejecutar un Diseño de Aprendizaje, que implique un seguimiento complejo de la actividad de los alumnos. En principio, solo dispone de una interfaz muy simple, pero facilita que otros desarrolladores puedan añadir una variedad de interfaces para crear diferentes reproductores especializados, como es el caso de Sled, que posibilita la ejecución de Unidades de Aprendizaje directamente con Internet Explorer. El proyecto Reload también contempla un visualizador IMS-LD, aunque este es autónomo.

Edubox es un reproductor de EML [41], desarrollado por Perot Systems Netherlands, utilizado por la OUNL para distribuir sus cursos en línea. OUNL ha alcanzado un acuerdo con Blackboard Inc para incorporar Edubox a su sistema, adaptándolo para gestionar Diseño de Aprendizaje IMS. Otros reproductores son SLeD Player [42], Reload LD Player [5] y Coppercore Player [3].

Por otra parte, diferentes editores de Diseño de Aprendizaje están disponibles en la actualidad. El editor de código abierto Reload LD Editor permite editar los tres niveles de IMS-LD mediante un sistema de pestañas.

Otros editores incluyen Educreator un editor actualmente en proceso de desarrollo en OUNL. El proyecto Alfanet Project [43] ha desarrollado un editor basado en Groove. También OUNL está trabajando en CopperAuthor [4], un editor para IMS-LD, y en un editor visual en el proyecto TENCompetence [44]. Por su parte, Yongwu Miao ha desarrollado Cosmos [45], un editor técnico con funciones avanzadas como la copia profunda de estructuras, que permite no únicamente copiar y pegar la estructura en sí, sino todos los elementos que se encuentren relacionados con ella.

Todos estos editores requieren que los autores tengan un buen conocimiento de la especificación, pero otros editores de alto nivel que no tienen esta exigencia están en la actualidad en proceso de desarrollo. Por ejemplo el sistema MOT+ [8] se está ampliando para incorporar funciones que permitan exportar en formato Diseño de Aprendizaje. Otro editor visual es ASK LDT [46] de la Universidad de Piraeus.

El sistema LAMS [13] dispone de una interfaz, arrastrar y soltar, muy fácil de usar que permite combinar diferentes actividades de aprendizaje dentro de la estructura de una lección. Actualmente, cumple con la especificación Diseño de Aprendizaje nivel A. Moodle [10] y la OUNL por su parte están trabajando en la importación, exportación y ejecución de Unidades de Aprendizaje y cursos. Tienen previsto el lanzamiento de la primera fase de integración a lo largo del 2008.

Por último, existen diversos motores de ejecución. Con el ya mencionado Coppercore, otros motores ya ejecutan los tres niveles de la especificación con distinto éxito: .LRN [14], netUniversité [9] y Athabasca (under development). De todos ellos .LRN (DotLRN-Project, 2006) es el que mantiene mayor nivel de desarrollo y mantenimiento encontrándose actualmente en fase de pruebas.

1.8.6. Grupos y proyectos de investigación sobre IMS Learning Design

Desde la publicación de las primeras especificaciones sobre *eLearning* hace más de una década el número de grupos de investigación y proyectos asociados dedicados a desarrollarlas, analizarlas y explotarlas ha crecido considerablemente. Más de treinta grupos y proyectos de investigación internacionales en torno a IMS Learning Design o directamente relacionados con ella muestran un panorama alentador para esta joven especificación. Desde el motor de ejecución Coppercore hasta los visualizadores de Reload o Sled pasando por la media docena de editores desde un punto de vista tecnológico se puede comprobar que IMS Learning Design está respaldada y despierta el interés fundamentalmente del sector académico (Open University of The Netherlands, Open University of the United Kingdom, Universities of Duisburg, Piraeus, Valladolid, Vigo...) pero también del sector comercial (eLive, 8Lem...). No obstante, y además de este interés, hay tres factores clave que muestran su impacto y relevancia.

Primero, el número de aplicaciones inspiradas en IMS-LD o que soportan la exportación de sus paquetes de información en paquetes IMS-LD. Los proyectos actuales centrados exclusivamente en IMS-LD (CopperCore, Reload LD Editor y LD Player, netUniversité, CopperAuthor...) se ven reforzados por otras aplicaciones sobre *eLearning* que comenzaron inspiradas en IMS-LD o que nacieron antes y que han considerado la migración o incorporación de IMS-LD en su modelo conceptual (LAMS, ASK, MOT+...). Dentro del reciente movimiento sobre estandarización del *eLearning* parece que el enfoque de IMS-LD, a su vez basado en el Educational Modeling Language (EML), es fuente de inspiración y desarrollo, pero también de contraste y confrontación, utilizándose para depurar y mejorar una aproximación a la generación y uso de escenarios de aprendizaje.

Segundo, y quizá más decisivo, el número de comunidades virtuales que surgen en torno a este tema es amplio y continúa incrementándose [15]. Generalmente auspiciadas por proyectos financiados por organismos oficiales (UNFOLD, Ladie, Lornet, iClass...) pero también con alguna representación espontánea proveniente de foros de discusión ya activos (Moodle), los temas recurrentes del *eLearning* nutren los foros de estas comunidades, impulsando el debate crítico sobre características, utilidad y funcionalidades de las especificaciones, sus fundamentos teóricos y las aplicaciones informáticas relacionadas. Estas comunidades representan el interés generalizado y recurrente de los grupos habituales de usuarios finales (fundamentalmente profesores, proveedores de contenidos y diseñadores de aprendizaje, pero también desarrolladores de sistemas e investigadores) en mejorar su formación online y adaptar la formación presencial a plataformas telemáticas rentabilizando al máximo el tiempo y el esfuerzo requeridos y logrando unidades fácilmente interoperables y reutilizables.

Tercero y último, la internacionalización. Aunque IMS nació en Estados Unidos, como SCORM, su presencia en Europa es cada vez mayor y el interés demostrado por países como Holanda, España, Reino Unido, Portugal, Bulgaria, Francia, Italia, Alemania y muchos otros es amplio. Del mismo modo, y fuera de Europa, Canadá, Australia, Rusia, Brasil o Marruecos representan sólo una pequeña muestra de la extensión y diversidad de la especificación.

Si juntamos los tres factores, aplicaciones IMS-LD, comunidades virtuales en activo y dispersión geográfica, contamos con un núcleo de actuación fuerte, respaldado por el mundo académico y por los usuarios potenciales con diversos emplazamientos.

Presentamos a continuación un listado alfabético no exhaustivo que muestra el panorama actual de proyectos y grupos de investigación, basado en un estudio previo realizado por el autor entre 2005 y 2006 [15; 47].

1 .LRN (www.dotLRN.org, Internacional)

Desarrollo de código abierto para aprendizaje e investigación, ampliamente extendido (más de medio millón de usuarios y traducciones a ochenta idiomas) y fácilmente instalable y configurable. Originalmente desarrollado por el MIT (Boston, EEUU) aunque no soporta especificaciones de *eLearning* se basa en los mismos conceptos para generación de estructuras de aprendizaje. Oficialmente permite la aplicación de cualquier tipo de pedagogía y, al igual que Moodle fomenta la adaptación y la relación comunitaria virtual para su implementación y soporte.

2 ACETS (www.acets.ac.uk, Reino Unido)

Este proyecto colaborativo desarrolla y evalúa procesos para implementar el uso de objetos de aprendizaje reutilizables en el mundo médico, fundalmentalmente en el gremio de los doctores. Aunque los alumnos de Medicina son el principal objetivo, también se pretende que los profesores utilicen estos objetos de aprendizaje para retro-alimentarse y aplicarlos en sus clases. Se centra, por tanto, en lograr procesos más sencillos y rápidos, así como más satisfactorios intelectual y educativamente hablando, en la utilización de objetos de aprendizaje. ACETS es un proyecto financiado por JISC y utiliza la generación de taxonomías educativas y su aplicación en una variedad de escenarios para lograr procedimientos estándar que sean fácilmente exportables y utilizables por otros objetos de aprendizaje o escenarios.

3 Advanced e-Services for the Knowledge Society (ASK) (www.certh.gr/&en/home/index.htm, Grecia)

Dentro del Informatics and Telematics Institute (University of Piraeus), Demetrios Sampson dirige el ASK, unidad de investigación que se centra en la creación de software educativo basado en estándares y metadatos. Desarrolla un producto denominado ASK LDT (www.ask.iti.gr) que permite la creación de Unidades de Aprendizaje bajo la especificación IMS Learning Design y el ASK LOM-RM que gestiona repositorios de objetos de aprendizaje.

4 Cepiah (www.hds.utc.fr/cepiah, Francia)

Grupo de investigación ubicado en la Université de Technologie de Compiègne - Centre de Recherches de Royallieu y centrado en el desarrollo de módulos que permiten la mejora y desarrollo del diseño, evaluación y modelos de enseñanza dentro de los sitios web centrados en formación. Con el objeto de ayudar a los profesores sin bagage técnico, se ha creado netUniversité, compuesto por un editor, un generador de sitios web y un módulo de administración. De esta manera, se pretende abarcar todo el ciclo de generación de escenarios de aprendizaje online, desde la concepción hasta la ejecución real en clase.

5 CETIS (www.cetis.ac.uk, Reino Unido)

Representa la educación secundaria y universitaria en los organismos de estándares educativos, tales como IMS, CEN/ISSS o IEEE LTSC. Se encarga de asesorar a estas instituciones sobre las implicaciones estratégicas, técnicas y pedagógicas de la adopción de estándares en tecnología educativa. También asesora técnicamente a los diversos projectos y programas de JISC, the Joint Information Systems Committee of the Higher and Further Education Funding Councils y a los programas sobre *eLearning* de la Unión Europea. CETIS trabaja distribuido en diversos grupos en torno a diversas especificaciones, incluyendo metadata, contenido educativo, cuestionarios, accesibilidad y otros. CETIS está financiado por JISC y gestionado por la Universidad de Bolton.

6 DialogPlus (www.dialogplus.org, Reino Unido y EEUU)

"Digital Libraries in Support of Innovative Approaches to Learning and Teaching in Geography" es un proyecto financiado por JISC (ver referencia en esta misma sección) que proporciona una estructura de información distribuida para que los graduados y post-graduados de titulaciones sobre Geografía puedan compartir recursos y enriquecerse mutuamente a través del aprendizaje colaborativo en línea y diversas aplicaciones de código abierto centradas en la generación de bibliotecas de recursos ad hoc.

7 Educational Technology Expertice Centre, OTEC (www.ou.nl/eCache/DEF/5/071.html, Holanda)

Dentro de la Open University of The Netherlands este instituto de investigación se centra en la tecnología educativa (evaluación, cursos, retroalimentación, navegación). Más concretamente el Departamento de Desarrollo, dirigido por Rob Koper, desarrolla proyectos centrados en estándares. Fue el creador del Educational Modeling Language que sirvió como base para la implementación de la especificación IMS Learning Design, base de gran parte de la discusión actual sobre estándares en *eLearning*. Actualmente, además de proyectos de investigación sobre navegación (ROMA, http://www.ou.nl/eCache/DEF/13/374.html), comunidades virtuales con Learning Network for Learning Design [6], aprendizaje adaptativo y retroalimentación con Planning Educational Task [48], desarrollo de competencias con TENCompetence [44], investigación sobre personalización y aprendizaje profesional con ProLearn [49] y un largo etcétera, trabaja en la generación de motores de ejecución para especificaciones (Coppercore [3]), editores (Copperauthor [50]), entornos virtuales de aprendizaje (Alfanet [43]), navegadores (Sled [42]), además de colaborar con casi todos los grupos de investigación anteriormente mencionados en esta página. Paralelamente, mantiene grupos de depuración y ampliación de especificaciones IMS (Content Packaging, Learning Design, QTI…), interoperabilidad (con SCORM, Moodle y LAMS), desarrolla una amplia labor divulgativa en proyectos (UNFOLD, Scope), conferencias (Online Educa Berlin, Online Educa Madrid, SIGOSSE, ICALT…), publicaciones (JCAL, ILE, JIME, ET&S…), cursos y talleres (Campus Virtual, Eucen…), consorcios oficiales de regulación (IMS, IEEE…) y un largo etcétera. Actualmente, constituye uno de los principales centros de investigación, desarrollo, debate y divulgación sobre tecnología educativa.

8 Edusource (http://www.edusource.ca, Canadá)

Proyecto colaborativo que pretendía (finalizó en 2004) generar una red repositorios interconectados e interoperables sobre objetos de aprendizaje a lo largo de Canadá. Su objetivo también era liderar el desarrollo de software, sistemas, protocolos y prácticas que permitieran explotar y mantener dicha red. Como proyecto bilingüe (inglés y francés) que permite la comunicación y el trabajo colaborativo entre todas las provincias francesas, precisaba de una conexión de banda ancha a Internet con capacidad suficiente para envíos masivos de información.

9 Elive LD Suite (www.elive-ld.com, Alemania)

Desarrollada por cogito GmbH esta aplicación permite la edición, documentación y optimización de escenarios de aprendizaje de manera gráfica. Está basada en la especificación IMS Learning Design y permitirá exportar Unidades de Aprendizaje Level A. El objetivo final de eLive es proporcionar una herramienta visual que permita generar mapas de conceptos, estructuras y métodos, acorde con el modelo pedagógico seleccionado por el diseñador de aprendizaje o profesor.

10 Grup de Tecnologies Interactives (www.tecn.upf.es/gti, España)

Perteneciente a la Universidad Pompeu Fabra (España) y dirigido por Josep Blat este grupo centra su investigación en diferenctes aspectos sobre interactividad multimedia e Internet. Entre otras áreas se encuentra el desarrollo de software y sevicios web, la consultoría de usabilidad, las aplicaciones de dispositivos móviles, los juegos y el *eLearning*. Trabaja en los proyectos UNFOLD y SCOPE además de una docena más de iniciativas financiadas por la Unión Europea.

11 Iclass (www.iclass.info, Europa)

"Intelligent cognitive-based open learning system for schools" es un proyecto que pretende desarrollar un sistema y un entornos abiertos de aprendizaje basado en inteligencia cognitiva, con el objeto de satisfacer las necesidades de alumnos individuales. El producto a obtener será una aplicación basada en una arquitectura ontológica de almacenamiento y recuperación de información que permita generar dinámicamente objetos individuales de aprendizaje. Para ello trabaja con una estructura de comunidad virtual orientada por objetivos y por entornos comunes de trabajo colaborativo.

12 Institute of Educational Technology, IET (http://iet.open.ac.uk, Reino Unido)

Dentro de la Open University of The United Kingdom, asesora sobre la utilización de las tecnologías modernas en la optimización del aprendizaje a distancia y online. Colabora con más de cincuenta proyectos nacionales e internacionales sobre tecnología educativa, investigación sobre aprendizaje e *eLearning*. Gracias a su inclusión dentro de la Open University lleva a cabo experimentos y desarrollos con alumnos/as que permiten un análisis de primera mano sobre casos reales, lo que revierte una mejora y depuración de los métodos y sistemas utilizados. Participa activamente en proyectos y comunidades virtuales sobre estándares de *eLearning* (UNFOLD) y en la generación de aplicaciones de aprendizaje (UoL) basadas en estándares, como es el caso del visualizador SLED de Unidades de Aprendizaje desarrolladas en IMS Learning Design, desarrollado al amparo del motor Coppercore, y soportando UoLs de nivel A. También es el caso del proyecto ELF Toolking (www.*eLearning*.ac.uk/news_folder/phase2toolkits), en el que trabajan Alex Little y Patrick McAndrew, que pretende lograr una integración de especificaciones IMS (LD y QTI principalmente) o del proyecto Demonstrator (www.*eLearning*.ac.uk/news_folder/toolkitdemonstrators), que pretende mostrar usos y características de la especificación IMS Learning Design a través de ejemplos concretos que puedan ser reutilizados.

13 JISC (www.jisc.ac.uk, Reino Unido)

The Joint Information Systems Committee of the Higher and Further Education Funding Councils (JISC) da apoyo a los sistemas educativos de secundaria y universitarios asesorando sobre las oportunidades y consecuencias de utilizar tecnologías de comunicación e información en la enseñanza, el aprendizaje, la investigación y la gestión administrativa. JISC trabaja mediante un comité de expertos (directores, gestores, académicos y tecnólogos) de educación secundaria y superior y proporciona nuevos entornos educativos, acceso a recursos electrónicos, redes de aprendizaje, guías para cambio institucional, asesoría y soporte regional a colegios.

14 Laboratorio DEI, proyecto CASLO (http://caslo.dei.inf.uc3m.es, España)

El DEI, perteneciente a la Universidad Carlos III (España), desarrolla el proyecto CASLO (Collaborative Annotation Service for Learning) dirigido por Juan Manuel Dodero. CASLO permite compartir la autoría sobre recursos de aprendizaje, habilitando secciones de código XML anotadas, que pueden ser comentadas y modificadas por el grupo de trabajo desde distintos puntos de acceso (al estilo de un sistema wiki) dentro de un tiempo determinado para, finalmente, proceder a la incorporación y validación consensuada del contenido final. El sistema puede trabajar con cualquier recurso basado en XML, como objetos LOM o IMS Learning Design.

15 Laboratory for Ontological Research (LORE) (http://lore.iat.sfu.ca, Canadá)

Perteneciente a la Simon Fraser University (Canada) y dirigido por Griff Richards, este grupo de investigación centra sus esfuerzos en la gestión de conocimiento y más concretamente en ontologías, interoperabilidad y razonamiento. Modelan systemas de aprendizaje online, construyen sistemas distribuidos e interoperables, entornos adaptativos y compartición de conocimiento. Su última investigación (el ECL Interoperability Center) desarrolla una infraestructura para repositorio de objetos de aprendizaje, coleccionando información técnica, componentes y servicios sobre interoperabilidad.

16 Labset, University of Liège (www.labset.net, Bélgica)

Ha desarrollado e implanta la metodología 8LEM, dirigida por Dominique Verpoorten, que se basa en la generación de más de ochenta escenarios pedagógicos típicos que se adaptan y personalizan según las necesidades concretas de cada cliente. Completamente centrado en facilitar al profesorado la adaptación de metodologías de enseñanza al medio online, parte de la creación de plantillas que identifican procesos de aprendizaje reales que son utilizados de manera inmediata por el profesor, sin necesidad de ningún conocimiento tecnológico previo.

17 Ladie (www.elframework.org/refmodels/ladie, Reino Unido)

Este proyecto fundado por JISC (ver referencia en esta misma sección) desarrolla un modelo de referencia que fundamenta el diseño y construcción de actividades de aprendizaje, así como el descubrimiento, la especificación, la secuenciación y el empaquetamiento de contenido. Del mismo modo, también sustenta los entornos donde las actividades formativas se llevan a cabo y la ejecución de las actividades formativas en sí mismas. El proyecto LADIE está relacionado intimamente con DialogPlus con el objetivo de describir y reconocer las necesidades tecnológicas asociadas a este modelo pedagógico de referencia.

18 LICEf, University of Quebec (www.licef.teluq.uquebec.ca/francais/real/mot.htm, Canadá)

Este grupo de investigación, dirigido por Gilbert Paquette, desarrolla desde antes de la aparición de los estándares en *eLearning* una metodología de creación de materiales pedagógicos formalizada en la aplicación MOT+, y que tiene grandes similitudes con IMS Learning Design, aunque no es completamente compatible. Se basa en la representación gráfica de cualquier metodología didáctica empleada para generar Unidades de Aprendizaje. Al igual que ocurre con ASK LDT las posibilidades de definición gráfica no permiten la importanción de recursos desarrollados con otras aplicaciones, aunque sí la exportación con un alto porcentaje de satisfacción.

19 Lornet (www.lornet.org, Canadá)

Centrado en la reusabilidad de objetos de aprendizaje diseminados a través de Internet, Lornet trata de definir los mecanismos de interoperabilidad que permitan localizar los elementos buscados y permitir adaptarlos al contexto preciso del usuario final. Para ello se pretende construir uno o varios prototipos centrados en la agregación de repositorios de objetos de aprendizaje multi-idioma y multimedia relacionados entre sí mediante metadatos y un sistema ontólogico de almacenamiento y recuperación de recuperación. Lornet utiliza las premisas de ciertas especificaciones centradas en *eLearning* (fundamentalmente de IMS Global Consortium) para realizar un análisis crítico de las mismas y complementarlas.

20 LT3 Centre (http://lt3.uwaterloo.ca, Canadá)

Perteneciente a la University of Waterloo (Canadá) y dirigido por Liwana Bringelson, desarrolla y comparte diseños instructivos innovadores, diseña y apoya comunidades virtuales de aprendizaje no estructurado, crea objetos de aprendizaje y colabora en repositorios, mide y analiza impactos de la tecnología en la enseñanza y en el aprendizaje. Concretamente, desarrolla el modelo T5 (tareas, tutorización, trabajo en equipo, recursos y herramientas) de aprendizaje y enseñanza, que predica el aprendizaje colaborativo entre los diversos miembros implicados en un procedo educativo. Esta teoría respalda la aplicación LearningMapR que, a su vez, trabaja sobre el concepto de plantillas de aprendizaje y Unidades de Aprendizaje (UoL), es decir, elementos reusables que definen casos de estudio y metodologías pedagógicas y que se ofrecen a la comunidad para su uso y adaptación contextual concretas.

21 Macquarie University and LAMS Foundation (www.lamsinternational.com, Australia)

Grupo de trabajo que desarrolla LAMS, aplicación académica dirigida por James Dalziel y centrada en el enfoque pedagógico de la construcción de materiales educativos, más que en la creación técnica. Los profesores suelen apreciar la sencillez para secuenciar lecciones y cursor y para diseñar estructuras de aprendizaje. Inspirada en la especificación IMS Learning Design, la próxima versión de LAMS será de código abierto y trabajará sobre la interoperabilidad de sus objetos de aprendizaje con otros sistemas y motores de ejecución.

22 Pool (http://www.edusplash.net, Canadá)

El Portal for Online Objects in Learning es un consorcio de diversas organizaciones, públicas y privadas, del sector educativo que desarrollan repositorios o bases de datos de objetos de aprendizaje. Para ello se trabaja tanto con la definición de arquitecturas locales basadas en metadatos (en concreto, el estándar IEEE LOM), como la realización de productos software y hardware que permitan alimentar dichas estructuras con contenido real. El producto que concreta todos los esfuerzos de esta comunidad virtual de trabajo colaborativo en torno a los objetos de aprendizaje y la categorización es la aplicación Splash, que en su última versión permite un vínculo con Edusorce. Finalizado en 2002.

23 Projecto Moodle (http://moodle.org, Australia)

Modular Object-Oriented Dynamic Learning Environment, sistema de gestión de cursos (CMS) de código abierto, desarrollado en lenguaje PHP, es un projecto originalmente particular fundado por Martin Dougiamas que se fundamenta en el constructivismo social (el individuo aprende a medida que interactúa con el entorno y con otros) de una comunidad educativa virtual muy extensa alrededor del mundo. Se desarrolla bajo gestión de licencia pública GNU (Free Software Foundation, 1989), lo que permite a cualquier interesado modificar y adaptar el código de la aplicación para a sus propios intereses siempre y cuando extienda dicha licencia a su creación.

Moodle está traducido a más de cincuenta idiomas y su comunidad virtual es una de las más activas dentro de los educadores y diseñadores de aprendizaje. Actualmente prepara la interoperabilidad entre el sistema LAMS (ver referencia en esta misma página) y la especificación de *eLearning* IMS Learning Design.

24 R2R (http://commons.ucalgary.ca/weblogs/learningdesign, Canadá)

Repository to Reality (R2R) se fundamenta en el trabajo con comunidades virtuales de profesores, estudiantes, pedagogos, desarrolladores e investigadores en torno a la especificación de *eLearning* IMS Learning Design (IMS-LD). Sub-grupos de interés en recomendaciones, herramientas y uso trabajan internamente y de manera descentralizada para proporcionar una base de conocimiento a investigadores y desarrolladores de la especificación, con el objeto de aplicarla y configurarla contextualmente.

Diversas universidades canadienses investigan sobre implicaciones técnicas y pedagógicas en la implementación de IMS-LD, desarrollan un vocabulario limitado y escriben una serie de informes sobre los retos de implementar la especificación según el contexto.

25 Reload Project (www.reload.ac.uk, Reino Unido)

Gestionado por la Universidad de Bolton, dirigido por Oleg Liber y financiado por JISC (ver referencia en esta misma sección) Reload desarrolla editores de código abierto, visualizadores y entornos virtuales basados en estándares (IMS Learning Design, IMS Content Packaging y SCORM) con el objetivo de crear y ejecutar unidades y objetos de aprendizaje. Actualmente el editor de IMS-CP está completamente activo, así como el editor de SCORM. En referencia a IMS-LD, el visualizador no es un player completo, sino que basado en el motor Coppercore (Vogten y Martens, 2004) desarrollado por la Open Univesity of The Netherlands, ejecuta de manera sencilla y mediante usuarios paja las Unidades de Aprendizaje diseñadas previamente. El editor de IMS-LD, por su parte, permite trabajar con los tres niveles de la especificación, aunque el nivel C no ha sido probado todavía.

26 Salerno University (www.unisa.it, Italia)

Participa en el proyecto ELeGI de la UE, construyendo Unidades de Aprendizaje (UoLs) personalizadas para un determinado estudiante, según su estado cognitivo preciso y las preferencias personalizadas, la pedagogía utilizada, la retroalimentación del usuario. Reuniendo toda esta información y administrándola mediante agentes informáticos, el proceso de personalización en el proceso de aprendizaje es máximo. Actualmente se trabaja en la posibilidad de implementar toda la teoría utilizando diversos estándares, IMS Learning Design entre ellos, potenciando un documento base o manifiesto que refleje adecuadamente la metodología.

27 SCOPE (http://www.tecn.upf.es/scope/showcase/, Europa)

Aunque el proyecto SCOPE finalizó el 30 de Junio de 2003, sus enlaces continuan activos y son fuente de consulta y comunicación. Mantiene una red de comunidades virtuales en diversos países de Europa con el objetivo de reutilizar derechos de propiedad intelectual científicos y convertirlos en objetos de aprendizaje y productos de alto valor. Para ello, se hacen disponibles contenidos mediante un servicio de suscripción web, se establecen mecanismos de búsqueda y recuperación de información multilingüe y se reutilizan contenidos particularizados para los profesionales médicos.

28 Telcert (www.opengroup.org/telcert, Europa)

Projecto de desarrollo e investigación amparado en el Sexto Programa Marco de la Unión Europea, está formado por un consorcio de proveedores de contenidos, investigación y organizaciones industriales y desarrolla aplicaciones informáticas para comprobar la interoperabilidad de de los sistemas *eLearning*, tanto en contenido como en tecnología. Una parte importante de su trabajo es generar perfiles de interoperabilidad y reusabilidad que permitan ser adoptados por las aplicaciones informáticas y las herramientas de re-ingeniería.

29 The Open Group (www.opengroup.org, Reino Unido)

Consorcio comercial y tecnológicamente neutral que proporciona acceso a información coordinada entre diferentes proyectos y aplicaciones basados en estándares abiertos y en interoperabilidad. Trabaja con clientes, proveedores y consorcios para detectar, comprender y solucionar necesidades del sector. También facilita la interoperabilidad entre las partes implicadas, la integración de diferentes especificaciones y los desarrollos de código abierto. Básicamente, es un foro de intercambio de ideas entre miembros de alto nivel, mezclando el mundo académico y el empresarial, centrados en tecnología educativa y en *eLearning* standards.

30 UNFOLD (www.unfold-project.net, Europa)

Understanding New Frameworks of Learning Design (UNFOLD) es un proyecto del Sexto Programa Marco de la UE y se centra en la implementación y uso de estándares sobre *eLearning*, tanto para usuarios individuales como para actividades multi-usuario, y en aplicaciones de código abierto. Otros temas de interés son la interoperabilidad de los productos, contenidos y plataformas de aprendizaje online y los modelos pedagógicos soportados por ellas. Su dinámica de trabajo se centra en las comunidades virtuales o de práctica en torno a diversos temas específicos para profesores, escritores, pedagogos, investigadores y programadores.

El proyecto UNFOLD es clave para la divulgación de IMS-LD y el apoyo a la formación de nuevos grupos de investigación desde el año 2003. Finalizado en 2005.

31 Universidad de Valladolid (http://gsic.tel.uva.es/index.php?lang=es, España)

Este grupo de investigación, con sede en la Escuela Técnica Superior de Ingenieros de Telecomunicación de Valladolid, tiene como objetivo trabajar en los campos de los Sistemas Inteligentes, que incluyen redes neuronales y sistemas neuro-difusos, con especial interés en los modelos derivados de la Teoría de Resonancia Adaptativa (ART); los Sistemas Cooperativos, es decir en CSCW (Computer Supported Cooperative Work) con especial énfasis en CSCL (Computer Supported Collaborative Learning) y el proyecto TELL, financiado por la Unión Europea dentro de su programa *eLearning*.

El proyecto TELL se enfoca en la aproximación pedagógica y didáctica del aprendizaje colaborativo apoyado por ordenador/red (CSCL o NSCL). Es un esfuerzo metódico y sistemático para apoyar la comprensión de los procesos de aprendizaje que suceden en entornos CSCL mediante patrones de diseño, realizar un meta-estudio de métodos y herramientas que miden la efectividad de procesos CSCL, ofrecer y proponer métodos y herramientas - toolkits para educadores que quieren medir la eficiencia de actividades CSCL, ofrecer medios para la formación de los actores humanos involucrados (o de los que quieren involucrarse) en actividades de aprondizaje colaborativo y apoyar el diseño de nuevas herramientas tecnológicas efectivas para aprendizaje colaborativo.

32 Universidad de Vigo (http://www-gist.det.uvigo.es/~mcaeiro/thesis.html, España)

El Grupo de Ingeniería de Sistemas Telemáticos de la Universidad de Vigo (España) desarrolla una investigación sobre re-utilización e interoperabilidad de Unidades de Aprendizaje escritas en IMS Learning Design y generalizadas en cualquier lenguaje de modelado educativo o EML. Particularmente, proponen extensiones a la especificación intentando solventar algunas deficiencias que encuentran ante situaciones concretas de enseñanza. Paralelamente desarrollan una herramienta que valore la capacidad de modelado pedagógico de un EML y un editor que utilice las extensiones propuestas para demostrar la validez de las mismas y su aplicación real a los escenarios pedagógicos que no son completamente realizables con la especificación actual.

33 University of Duisburg (http://www.unfold-project.net/general_resources_folder/cosmos_tool.zip, Alemania)

Desarrolla un producto llamado Duisburg Collaborative Learning Authoring System que permite el diseño de Unidades de Aprendizaje mediante un sistema de notación propio XML, aportando elementos condicionales, propiedades y disparadores asociados a eventos concretos. La gran similitud con IMS Learning Design y la necesidad de asociarse con una especificación estándar ya operativa ha llevado a su creador, Young Wu Miao, a adaptarlo rápidamente hasta niveles B y C, creando COSMOS.

34 University Pierre et Marie-Curie Paris 6 (Francia)

Monique Baron dirige un grupo de investigación que desarrolla un editor y un simulador basado en las especificaciones Educational Modeling Language (OUNL, 2000) e IMS Learning Design. El objetivo principal es dotar a la comunidad educativa de una herramienta capaz de modelar unidades de enseñanza y utilizar objetos de aprendizaje de una manera sencilla y adaptada a la didáctica propia de cada profesor.

35 Alfanet (http://adenu.ia.uned.es/alfanet, Europa)

Proyecto finalizado en 2004, desarrolló un sistema de aprendizaje basado en IMS-LD. Una de sus aportaciones fue el diseño del motor de ejecución CopperCore (www.coppercore.org).

36 EU4ALL (www.eu4all-project.eu, Europa)

Integrated Project (IP) centrado en aprendizaje online para discapacitados. El subproyecto SP5 trabaja con IMS-LD y estructuras y escenarios de personalización basadas en guías pedagógicas y psicológicas.

37 Grapple (sin web todavía, Europa)

Proyecto recientemente aprobado en FP7, que desarrollará un sistema de aprendizaje adaptativo y que contará con un paquete de trabajo centrado en IMS-LD y personalización.

38 Kaleidoscope (www.noe-kaleidoscope.org, Europa)

No define ningún paquete específico para IMS-LD, aunque incorpora una red de investigadores que sí lo utilizan. Actúa por tanto como red social que se complementa con UNFOLD y ProLearn, a distinto nivel.

39 LN4LD (http://imsld.learningnetworks.org, Open University of The Netherlands)

Red de aprendizaje centrada en IMS-LD que ha sido utilizada por diversos proyectos (UNFOLD, ProLearn, TENCompetence, PET…) como vía de comunicación e impulso de la especificación y de investigaciones relacionadas.

40 LUISA (www.luisa-project.eu, Europa)

Centrado en definición de ontologías de aprendizaje. Una parte aborda la conexión con IMS-LD.

41 ProLearn (www.prolearn-project.org, Europa)

Red de excelencia centrada en aprendizaje profesional, con un paquete de trabajo (WP1) sobre aprendizaje adaptativo. En las dos últimas definiciones de trabajo (JPA3 y JPA4) la labor del WP1 se centra en la descripción y utilización de procesos de aprendizaje adaptativo utilizando IMS-LD. Queda reflejado en los informes D1.10 y D1.11, actualmente en revisión, y en los informes D1.08 y D1.09 ya aprobados por la Comisión Europea.

42 SUMA (sin web todavía, Plan Avanza, España)

Proyecto nacional que integra diversos servicios dentro de la capa OKI, entre ellos extensiones de Moodle y mapeo con IMS-LD. El subproyecto 4 se centra en sistemas adaptativos.

43 TENCompetence (www.tencompetence.org, Europa)

IntegrateD Project (IP) centrado en el desarrollo de competencias, incluye una parte de IMS-LD sobre un sistema visual de autoría basado en el muy técnico Reload LD Editor. Entrega de prototipo planeada para 2007.

1.8.7. **Listado de investigadores con trabajos recientes relacionados**

La siguiente lista de investigadores y la breve descripción del trabajo desarrollada por ellos no guarda una relación directa con la tesis, aunque ellos comparten o pueden compartir cierta inquietud sobre IMS-LD, integración de Unidades de Aprendizaje y/o aprendizaje adaptativo. A continuación, se describen las principales líneas actuales que, de alguna manera pero tangencialmente, tocan el objeto de esta tesis. Se incluye el nombre del investigador principal y la universidad donde desarrolla su actividad. En el plano nacional español (orden alfabético por apellido):

1. Amorim (Universidad de Santiago de Compostela) defendió en junio de 2007 una tesis doctoral centrada en el desarrollo ontológico de aprendizaje online, donde incluía una parte sobre IMS-LD
2. Boticario & Santos (UNED), trabajan actualmente en la arquitectura de un sistema de aprendizaje centrado en accesibilidad, que utilizará .LRN y que abordará la personalización de contenidos y procesos. Asimismo, llevan una línea de investigación sobre modelado de usuario en IMS-LD para desarrollo de sistemas de aprendizaje adaptativo
3. Caeiro (Universidad de Vigo) defendió en septiembre de 2007 una tesis doctoral centrada en los lenguajes de modelado educativo. Existe una parte teórica que aborda IMS-LD desde el modelado
4. Dodero et al. (Universidad Carlos III) llevan una línea de investigación sobre autoría no representacional de diseño instructivo y enfoque model-driven para reutilización de componentes. Gran parte de esta investigación es aplicable a IMS-LD
5. Escobedo, Delgado Kloos et al. (Universidad Carlos III) profundizan en la definición y utilización de .LRN. Esta fase de pruebas cuenta con una parte centrada en la ejecución de Unidades de Aprendizaje modeladas en IMS-LD para los tres niveles de la especificación
6. Fernández-Manjón (Universidad Complutense) trabaja sobre estándares eLearning y actualmente desarrolla un motor de QTI, así como una herramienta de autoría visual de IMS-LD (<e-LD>), que será evaluada en el contexto del proyecto europeo EU4ALL finales de 2007. Asimismo, está a punto de publicar un libro sobre avances en educación e informática donde hay un capítulo centrado en IMS-LD y adaptación, referenciado en el anteproyecto
7. Hernández-Leo (Universidad de Valladolid) defendió en julio de 2007 una tesis doctoral centrada parcialmente en la utilización de plantillas y en la reutilización de elementos de modelado con IMS-LD, así como en el aprendizaje colaborativo

8. Moreno-Ger y Sierra (Universidad Complutense) han publicado diversos artículos y Moreno-Ger defendió en diciembre de 2007 una tesis doctoral con una parte centrada en el desarrollo de un motor de aprendizaje para interpretación de juegos conversacionales. Existe una parte que trabaja la integración de dicho motor con IMS-LD

9. Sánchez & Lama (Universidad de Santiago de Compostela) llevan una línea de investigación sobre ontologías de aprendizaje y mapeado de diversas representaciones de modelado educativo, como LDL, MOT+, LAMS y IMS-LD

10. Zarraonandía (Universidad Carlos III) defendió en julio de 2007 una tesis doctoral centrada en la definición de un modelo, método y mecanismo de implementación de procesos de aprendizaje que puedan ser modificados en tiempo de ejecución por el instructor. Afronta una definición de un modelo de ejecución que permite diluir la diferencia entre momento de diseño y momento de ejecución. Aunque pudiera estar relacionado con este anteproyecto porque habla de modificación de procesos en tiempo real (un tipo específico de personalización) no trabaja sobre la especificación en sí sino con el diseño conceptual para EMLSs y la definición de un mecanismo de adaptación con repercusión en un motor de ejecución. IMS-LD se utiliza como un ejemplo de implementación junto con Moodle. Es decir, no reescribe o extiende la especificación sino que la usa como ejemplo de manera complementaria. Aunque únicamente aborda uno de los diversos tipos de adaptación existentes y no específicamente para IMS-LD, es un enfoque interesante

En el plano internacional existen ciertos trabajos sobre IMS-LD:

13. Botturi (Università Della Svizzera Italiana) acaba de publicar un libro sobre lenguajes visuales para diseño instructional. Hay dos capítulos que abordan la utilización del diseño visual para IMS-LD, referenciados en el anteproyecto

14. Cristea (Warwick University) trabaja sobre aprendizaje adaptativo centrado en LAOS y MOT. Desde inicios de 2006 intenta realizar junto con el autor un mapeado de LAOS con IMS-LD

15. DeBra (TU/e) trabaja sobre aprendizaje adaptativo en web con AHA!. Su punto de relación con IMS-LD era Cristea, que migró a Warwick University. Actualmente, es Project Leader de Grapple (financiado por FP7), proyecto que comenzó en enero de 2008 y que desarrolla un Work Package sobre IMS-LD y adaptación

16. Dougiamas (Moodle) busca el entendieminto con otros sistemas y estándares. Trabaja con IMS-CP, SCORM, LAMS y con IMS-LD

17. Giglione & Dalziel (LAMS) buscan el entendimiento con otros sistemas y estándares, entre ellos IMS-LD, para el que ya pueden exportar cursos de nivel A
18. Kravcik & Gasevic (Open University of The Netherlands & Simon Fraser University) investigan sobre web semántica y ontologías educativas. Parte de esta investigación trabaja con IMS-LD
19. Liber et al. (Bolton University) trabajan sobre un editor visual de IMS-LD basado en Reload LD Editor, que verá la luz dentro del proyecto TENCompetence
20. Miao (Open University of The Netherlands) trabaja sobre la representación de mecanismos de coordinación en IMS-LD para soportar aprendizaje grupal
21. Naeve et al. (KTH & Upsala University) trabajan sobre la definición de ontologías de aprendizaje y modelado conceptual. Guardan alguna relación con IMS-LD a través de los proyectos europeos ProLearn y Luisa
22. Paquette et al. (TeleUniversité) trabajan sobre el modelado con MOT+ y contemplan la exportación en distintos fases a IMS-LD. Conceptualmente guarda una cierta similitud con IMS-LD en la parte de autoría visual aunque necesita todavía mucho trabajo de aproximación
23. Sietze (Open University of United Kingdom) investiga sobre los diseños instructivos contextuales mediante la utilización de servicios web semánticos
24. Specht (Open University of The Netherlands) coordina un Work Package en el proyecto europeo ProLearn centrado en aprendizaje adaptativo. Una de sus contribuciones trabaja sobre cómo IMS-LD lleva o puede llevar a cabo dicha adaptación
25. Tattersall (QHN) ha publicado recientemente un artículo centrado en la utilización conjunta de IMS-LD y SCORM, y que se estudia como parte del análisis de esta tesis
26. Vignolet & Martel (University of Savoie) desarrollan el lenguaje de modelado LDL. De alguna manera, se intenta un entendimiento con IMS-LD

Algunos trabajos fruto de estas investigaciones se han publicado en el número especial del Journal of Interactive Media in Education (JIME) (http://jime.open.ac.uk/2007/01) titulado “Adaptation and IMS Learning Design”. También existen dos números especiales (uno en Computers in Human Behavior y otro en Simulation & Gaming) actualmente en proceso de edición en las que se encuentran algunos artículos relacionados bajo el proyecto conjunto denominado EGAEL (http://www.open.ou.nl/dbu/egael2007, http://www.danielburgos.eu), aunque el tema central de ambos números combina aprendizaje adaptativo y juegos (eGames).

Aunque existen otras iniciativas que referencian o utilizan IMS-LD en su trabajo (Alicante, Pompeu Fabra, Granada, Cartif, Minho, Laclo, Guatemala, Católica de Chile, Utrecht…) no han sido incluidas en esta lista ni en la tesis ya que, o utilizan de manera muy somera alguno de sus temas, o no realizan ninguna aportación que pueda ser referenciada todavía.

2. **Análisis de IMS Learning Design (IMS-LD)**

Esta sección comienza el núcleo de investigación de esta tesis, compuesto por tres partes, como son los aspectos generales de IMS LD, la integración y la adaptación de Unidades de Aprendizaje (UoLs). Fruto del estudio de esta sección se proporciona una serie de conclusiones parciales centradas en la especificación genérica (tales como estructura, esquema, diseño de aprendizaje, elementos y niveles) y que serán utilizados en secciones posteriores como base de su desarrollo para profundizar en la expresividad pedagógica de IMS-LD a través de la integración y la adaptación de UoLs.

2.1 . **IMS-LD: Definición y estructura**

En 2003, el IMS Global Learning Consortium Inc. publicó el Learning Design [51]. Esta especificación es una forma flexible de representar y codificar escenarios de aprendizaje para múltiples alumnos. Puede ayudar pensar en ella como una forma de crear planes de lecciones interoperables que pueden ser leídos por una aplicación denominada "player". El "player" puede encargarse de coordinar a los alumnos, profesores, recursos de aprendizaje y actividades a medida que el proceso de aprendizaje evoluciona.

De acuerdo a la especificación IMS-LD "*El concepto central de IMS-LD es que, independientemente de la aproximación pedagógica, una persona toma un rol en un proceso de aprendizaje-enseñanza, normalmente el rol de alumno o docente. En este rol la persona tiene que trabajar para conseguir ciertos resultados mediante la realización de actividades de aprendizaje o soporte más o menos estructuradas en un entorno*" [51]. Las características particulares de los roles que una persona puede tomar, las actividades a realizar y las características particulares del entorno definen un escenario de aprendizaje específico. Este escenario de aprendizaje puede ser representado en IMS Learning Design, donde se denomina una Unidad de Aprendizaje o Unit of Learning (UoL). La UoL puede ser ejecutada en algún sistema compatible Learning Design.

IMS-LD no ofrece ningún modelo o modelos pedagógicos concretos, sino que puede ser utilizado para definir prácticamente una serie ilimitada de escenarios educativos. A causa de esto se suele referenciar como un meta-modelo pedagógico. Algunas iniciativas anteriores de aprendizaje electrónico habían reclamado su neutralidad pedagógica. Learning Design no busca la neutralidad pedagógica, sino que trata de permitir que el *eLearning* sea consciente de la pedagogía [52; 53].

Learning Design fue desarrollado en un contexto de *eLearning*, pero no hay ninguna razón por la que las Units of Learning no puedan ser utilizadas en combinación de contextos de aprendizaje en línea y cara a cara, o de forma completa en aprendizaje cara a cara.

IMS-LD es el intento de avanzar diseñando para alumnos aislados en línea que están limitados a leer desde las pantallas, IMS-LD agrupa personas, actividades, recursos, flujos … en escenarios para alcanzar objetivos de aprendizaje. "*Se reconoce que el aprendizaje puede ocurrir sin objetos de aprendizaje, que el aprendizaje es diferente del consumo de contenido y que el aprendizaje surge de ser activo. Se reconoce, también, que el aprendizaje surge cuando los alumnos cooperan para resolver problemas.*" [54]. La cuestión principal no es crear contenido, sino crear actividades de aprendizaje estructuradas diseñadas para alcanzar objetivos de aprendizaje.

Por lo tanto, IMS apostó por una especificación centrada en el proceso de aprendizaje y no tanto en los contenidos finales, intentando asegurar la interoperabilidad de los módulos o Unidades de Aprendizaje (UoL) generados con ella. Mediante IMS-LD el profesor, pedagogo o escritor puede representar un escenario de aprendizaje sustentando en cualquier pedagogía, ya que es completamente neutro en este aspecto [54]. Es, por tanto, pedagógicamente flexible, lo que implica además una ejecución multiplataforma, independientemente del visualizador IMS-LD utilizado.

2.2 . **Metáfora de IMS-LD**

Learning Design utiliza la metáfora del teatro para ayudar a entender las Unidades de Aprendizaje. Una serie de actores representa una obra/ejecución/play; cada uno de ellos puede asumir un número de roles en diferentes momentos de la obra, en varios actos. Del mismo modo, en Learning Design, un estudiante puede asumir diferentes roles en diferentes etapas del proceso de aprendizaje. Al final de cada acto se produce una sincronización entre los participantes. El método define uno o varios plays. Cada play consta de uno o varios actos. Cada acto consta de uno o varios role-parts. Cada role-part define la actividad que realizará un determinado rol en un momento determinada. Esta actividad puede tener un entorno enlazado. La siguiente ilustración (Figure 7) muestra las dependencias entre los distintos componentes.

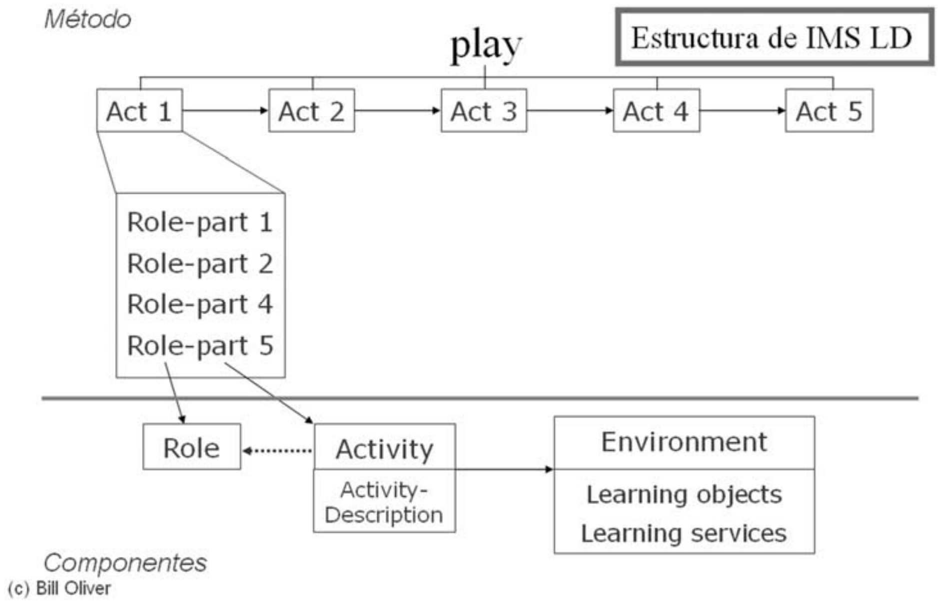


Figure 7. Diagrama de un método IMS-LD (Fuente: Olivier, 2003)

2.3 . Qué es una Unidad de Aprendizaje y cómo se utiliza

Una unidad de aprendizaje (o *Unit of Learning*, UoL) es [52] …"*una unidad de educación o formación completa y autónoma, curso, módulo o lección. La creación de una unidad de aprendizaje implica la creación de un diseño de aprendizaje y también la compilación de sus recursos asociados, bien como files contenidos en la unidad o como referencias web, incluyendo evaluaciones, materiales de aprendizaje e información para configurar el servicio de aprendizaje*".

Por lo tanto, es un fichero ZIP con:

- Un manifiesto XML, que describe el método, las ejecuciones, los actos, los roles, las actividades, los entornos, las propiedades, las condiciones y/o las notificaciones de la especificación Learning Design. También especifica los enlaces con los recursos asociados
- Ficheros opcionales de tipo HTML con contenido `webcontent`
- Ficheros opciones de tipo XML con referencias al manifiesto y los diversos elementos de IMS-LD
- Un conjunto de ficheros o recursos de cualquier tipo referenciados desde el manifiesto XML anterior

Si lo comparamos con otra especificación, IMS Content Packaging [19], esta última construye también paquetes de información con recursos siguiendo una serie de reglas, pero sin ningún método pedagógico subyacente. La diferencia, pues, entre IMS-LD y IMS Content Packaging es que IMS-LD añade a IMS Content Packaging una declaración completa bajo la etiqueta Organizations (donde se encuentra la formulación pedagógica del paquete) (Figure 8).

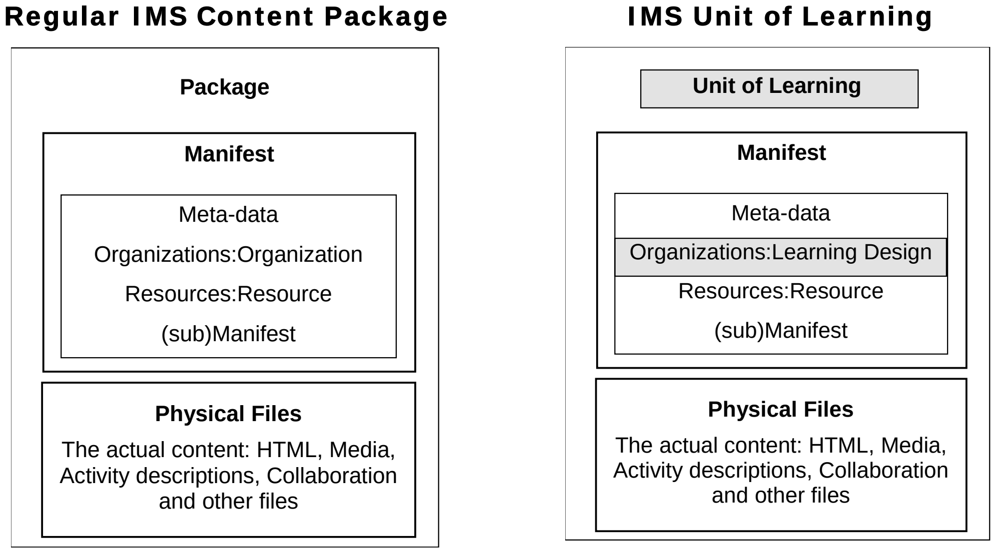


Figure 8. IMS Content Packaging vs. IMS Learning Design

Aparte, si comparamos contenidos creados en HTML con una estructura escrita para un manifiesto IMS-LD en XML podemos advertir que, además de las diferencias propias del lenguaje, hay otros puntos divergentes:

- Un manifiesto XML es un fichero único que apunta a recursos, mientas que un fichero HTML es un recurso en sí mismo, que también puede contener referencias a otros recursos
- Un manifiesto XML muestra el esqueleto y el método pedagógico de una Unidad de Aprendizaje, mientras que un sitio web HTML es únicamente un conjunto de páginas web enlazadas y estructuradas
- Un manifiesto XML puede proporcionar condiciones, propiedades y notificaciones, esto es un tipo de comportamiento dependiento de las acciones del usuario, mientras que un sitio web HTML es una fuente de información pasiva

En las dos siguientes ilustraciones (Figure 9, Figure 10) vemos la distribución de los distintos tipos de información dentro de un paquete de información IMS-LD: a) manifiesto, con la configuración del esqueleto básico y otros elementos; b) `imsldcontent`, con contenido XML y enlace con global elements, properties y monitoring services; c) `webcontent`, con contenido HTML; d) recursos en general de cualquier extensión (RTF, PDF, PPT, SWF...).

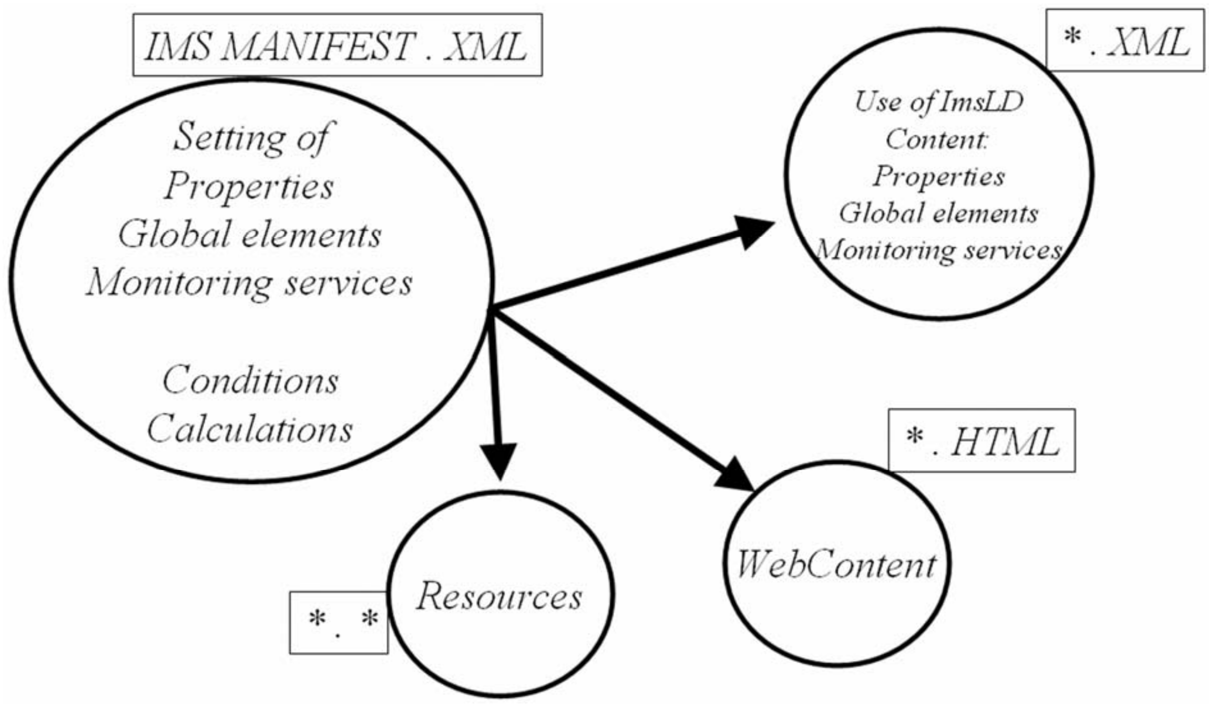


Figure 9. Tipos de contenido de un paquete de información IMS-LD

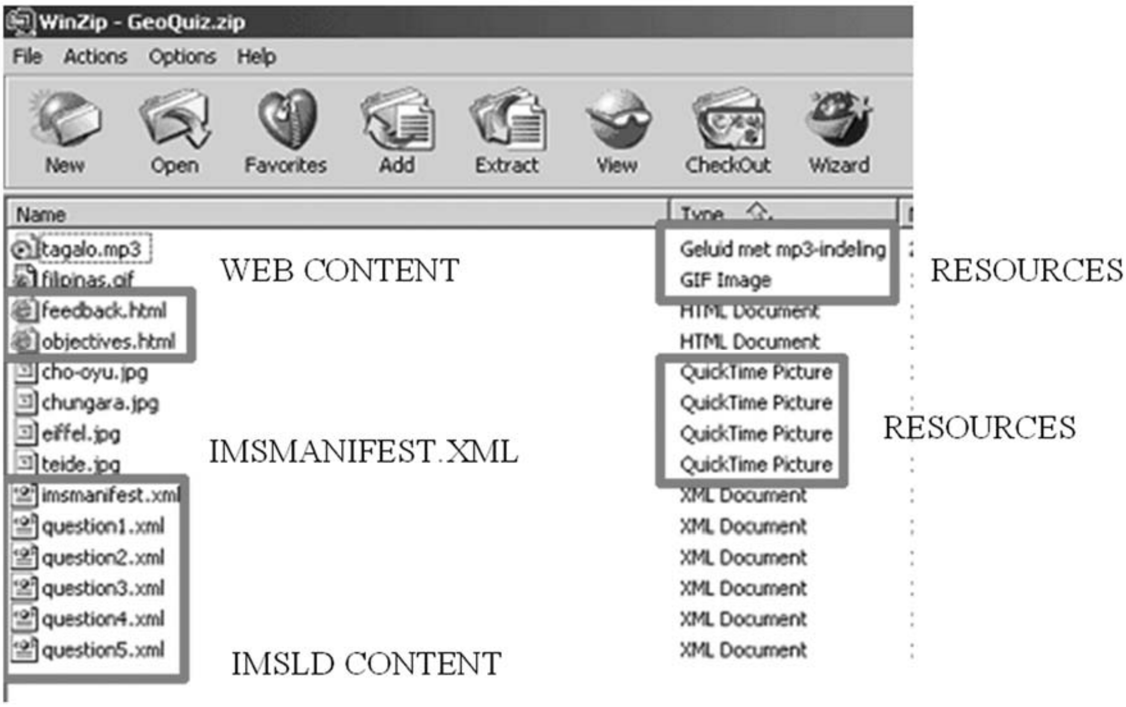


Figure 10. Contenido tipo de una unidad de aprendizaje en un fichero ZIP

Una vez que la unidad de aprendizaje se crea en el momento de diseño (o *design-time*) y el fichero ZIP es generado, se valida contra un motor de ejecución (ej. CopperCore) y se publica. Si todo es correcto, se almacena en la base de datos de este motor. Posteriormente, comienza el momento de ejecución (o *run-time*) y se comienza el proceso administrativo, en el que se activa una instancia de la UoL, se dan de alta los usuarios, y se asignan a roles determinados. La última parte del proceso es la ejecución con un visualizador. La siguiente ilustración (Figure 11) muestra estas etapas:

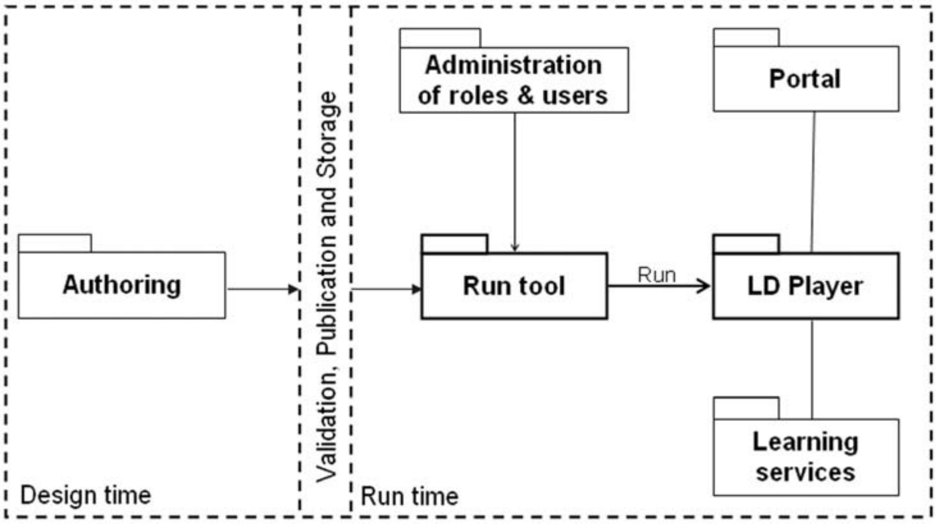


Figure 11. Etapas desde el momento de diseño a la ejecución de una UoL

En el momento de la visualización (con Reload LD Player, CopperCore Player, o Sled Player, por ejemplo) la ventana se encuentra dividida en tres áreas (Figure 12): 1) Árbol con el método, que muestra la estructura de los contenidos, siguiendo la metáfora del play, acto, estructura de actividades y actividades; 2) Área de visualización del contenido en sí, pudiendo ser de tipo `webcontent` o `imsldcontent`; 3) Área para los entornos complementarios, conectados a recursos o a servicios (ej. Comunicación y monitorización).

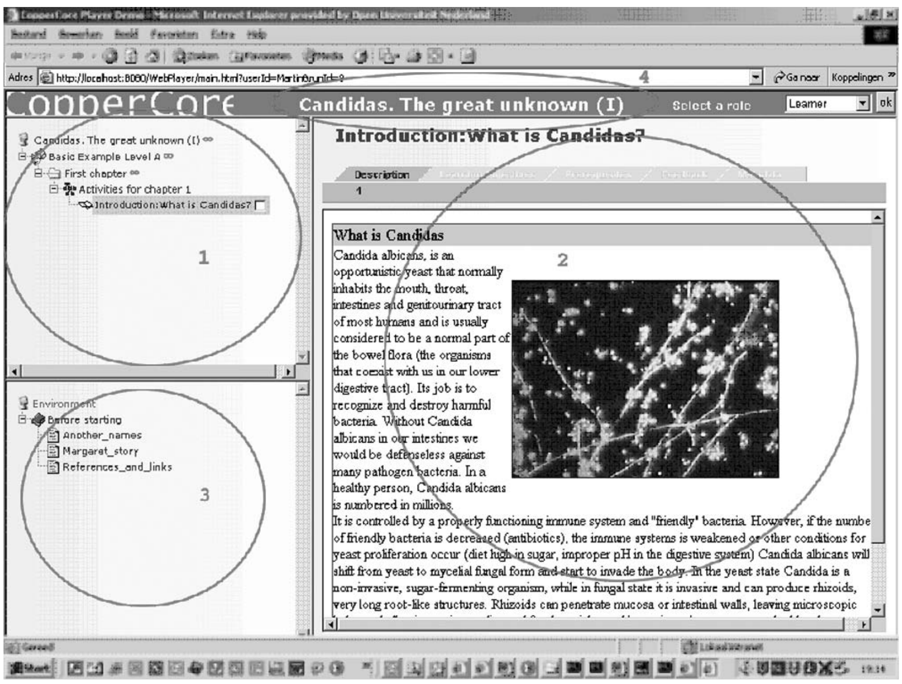


Figure 12. Áreas de un visualizador IMS-LD ejecutando una UoL

2.4 . Nacimiento y evolución de IMS-LD

El IMS Global Learning Consortium, que fue quién produjo el Learning Design, se creó en 1997 [22]. Ese mismo año, la Open University of The Netherlands (OUNL) tomó la decisión estratégica de hacer del *eLearning* una de las claves de su futuro desarrollo. Muchas pedagogías estaban ya en uso en la OUNL, y necesitaban soporte. Siguiendo esa premisa, la OUNL buscó un sistema de notación capaz de describir un amplio rango de modelos pedagógicos. Una vez descritos estos modelos, un reproductor los interpretaría del mismo modo que un navegador interpreta HTML.

Con este objetivo en mente, la OUNL desarrolló el Educational Modeling Language (Lenguaje de Modelado Educacional) [55], un meta-lenguaje usado para describir el proceso de aprendizaje y que soportaba diferentes enfoques del aprendizaje.

EML hizo que la OUNL se implicara en un programa interno de I+D de tres años, y tres ciclos de especificación-implementación-refinamiento. La primera versión resultó ser demasiado general porque describía muchas pedagogías, pero no con suficiente detalle. La segunda versión era demasiado específica, pues se limitaba a un rango particular de pedagogías. Finalmente, la tercera versión tuvo éxito al describir un amplio rango de pedagogías con suficiente detalle como para ser útil. EML v1.0 se publicó en Diciembre de 2000.

Hacia finales del año 2000, IMS había desarrollado especificaciones para describir contenidos y objetos educativos (Metadata), para realizar matrículas y devolver resultados (Enterprise), para transferir y mostrar contenidos (Content Packaging), para describir tests portables y devolver resultados (QTI), para describir estudiantes y su aprendizaje (LIP).

IMS reconocía, sin embargo, que era necesario ir más allá, y proporcionar una especificación para describir el proceso de aprendizaje en sí mismo. Era un objetivo ambicioso, que debía soportar diversos enfoques para el aprendizaje, basados en usuario único o multi-usuario y manteniendo la portabilidad, la reusabilidad y las capacidades de búsqueda. Por tanto, IMS necesitaba una descripción del diseño de más alto nivel, así que el trabajo empezó con el Learning Design (LD) Working Group (WG)

Después de trabajar en el problema durante algún tiempo, el Working Group decidió que sería más efectivo aprovechar una especificación existente que construir una nueva completamente. Esto llevó a la decisión de adoptar el EML de la OUNL, que fue aceptado en 2001 como una especificación de entrada sobre la que trabajar. La estructura detallada del EML cambió substancialmente en su transformación a IMS-LD, pero los conceptos del núcleo se mantuvieron igual.

IMS Learning Design fue aprobada como una Especificación Final IMS el 10 de Febrero de 2003. Como resultado, el EML ya no se mantiene ni actualiza, y la atención de la OUNL se centra en IMS-LD.

A pesar de que EML e IMS-LD comparten un objetivo y filosofía comunes, existen diferencias entre ambos. La siguiente tabla (Table 1) señala las diferencias.

Table 1. Diferencias entre EML e IMS-LD (Fuente: Tattersall and Koper, 2004)

EML	IMS-LD
Producido por la OUNL y puesto a disposición de la comunidad de la tecnología educativa.	Una especificación desarrollada y promovida por IMS.
Posee un modelo de contenido, que indica cómo deben ser estructurados los objetos de aprendizaje.	Sin modelo de contenido. Se recomienda el uso de XHTML, aunque soporta otros contenidos (p.ej. RTF).
Un único enfoque para desarrollar experiencias educativas.	Un marco de trabajo, integrado con IMS Content Packaging que integra IMS Meta Data, IMS Question and Test Interoperability, IMS Simple Sequencing y otras especificaciones IMS.

2.5 . **Cómo está estructurada la especificación**

La especificación IMS-LD comprende diversos documentos:

a) Un modelo conceptual que define los conceptos básicos y las relaciones dentro de Learning Design (conceptual model)

b) Un modelo de información que describe los elementos y los atributes que pueden ser definidos en IMS-LD de una manera detallada (information model)

c) Un conjunto de ficheros XSD (XML Schemas) en los que se basa la implementación del modelo de información

d) Una guía para la óptima implementación de la especificación (Best Practices and Implementation Guide)

e) Un conjunto de ejemplos y escenarios desarrollados en XML (binding document)

IMS-LD se divide a su vez en tres niveles de implementación (Figure 13):

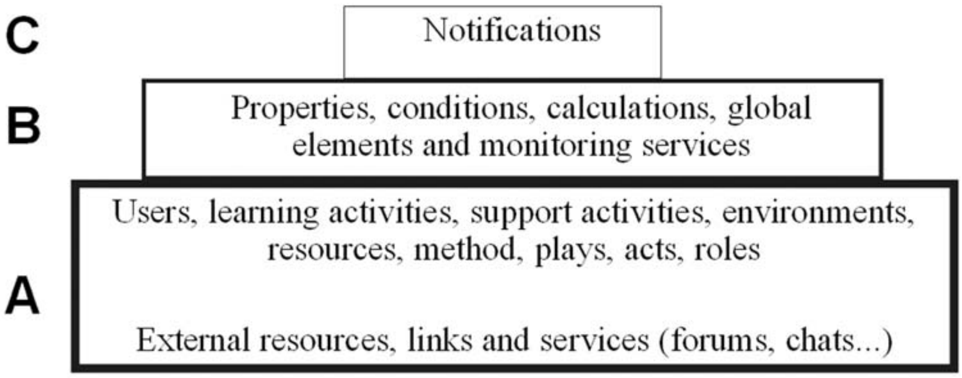


Figure 13. Niveles A, B y C en IMS-LD

- **Nivel A (Level A)**

Constituye la base y comprende la definición de usuarios, actividades de aprendizaje, actividades de soporte, entornos, recursos, método, ejecuciones (*plays*), actos, roles y la coordinación entre todos ellos, es decir, la expresividad pedagógica. Además los usuarios podrán utilizar recursos externos, enlaces web y diversos servicios (foros, chats...). Se pueden definir también las condiciones de finalización de las actividades, de la estructura, del acto, del play y del método. Estos distintos componentes estructurales de la especificación se muestran en la siguiente ilustración (Figure 14), que será detallada en el ejemplo de UoL de la próxima sección:

Title	Título
Learning objectives	Objetivos de aprendizaje
Pre-requisites	Prerrequisitos
Roles	Roles
Learning activities	Actividades de aprendizaje
How many, completion	Cuántas, fin
Supporting activities	Actividades de soporte
How many, completion	Cuántas, fin
Activity structure	Estructura de actividades
Method/Learning strategy	Método de aprendizaje
Finished?	Completado?
Resources	Recursos
Environments	Entornos

Figure 14. Estructura básica de un diseño creado con IMS-LD

- **Nivel B (Level B)**

A la base propocionada por el Nivel A añade propiedad, cálculos y condiciones, así como servicios de monitorización y elementos globales, lo que permite al usuario definir estructuras más complejas. Las propiedades almacenan información sobre personas (preferencias, resultados, información personal...), sobre un rol o sobre el diseño de aprendizaje en sí mismo. Si las propiedades son locales, se denominan internas y se mantienen únicamente durante la ejecución de una instancia (run). Si son globales, también llamadas externas, pueden ser consultadas y utilizadas en diferentes instancias y sus datos persisten a través de varias sesionies. El estado de las propiedades y de las condiciones puede modificar el flujo de trabajo e influir en el desarrollo de la Unidad de Aprendizaje (UoL). Constituye el nivel que aporta más flexibilidad a la hora de la representación didáctica ya que permite esconder y mostrar elementos, condicionar el flujo de aprendizaje, almacenar datos del usuario y la instancia, bien a nivel local y personal, bien a nivel global y compartido. De esta manera podemos incorporar cuestionarios, evaluaciones, cálculos numéricos, seguimiento de usuarios y un largo etcétera en el que profundizamos en el siguiente apartado.

- **Nivel C (Level C)**

Añade notificaciones al Nivel B. Las notificaciones se ejecutan automáticamente como respuesta a eventos que se originan en el proceso de aprendizaje. Por ejemplo, si un estudiante envía un trabajo opara ser evaluado, se podría enviar automáticamente un correo electrónico al profesor para informarle

La siguiente ilustración (Figure 15) muestra la arquitectura interna de los tres niveles con detalle de las dependencias entre ellos mismos y los diversos componentes:

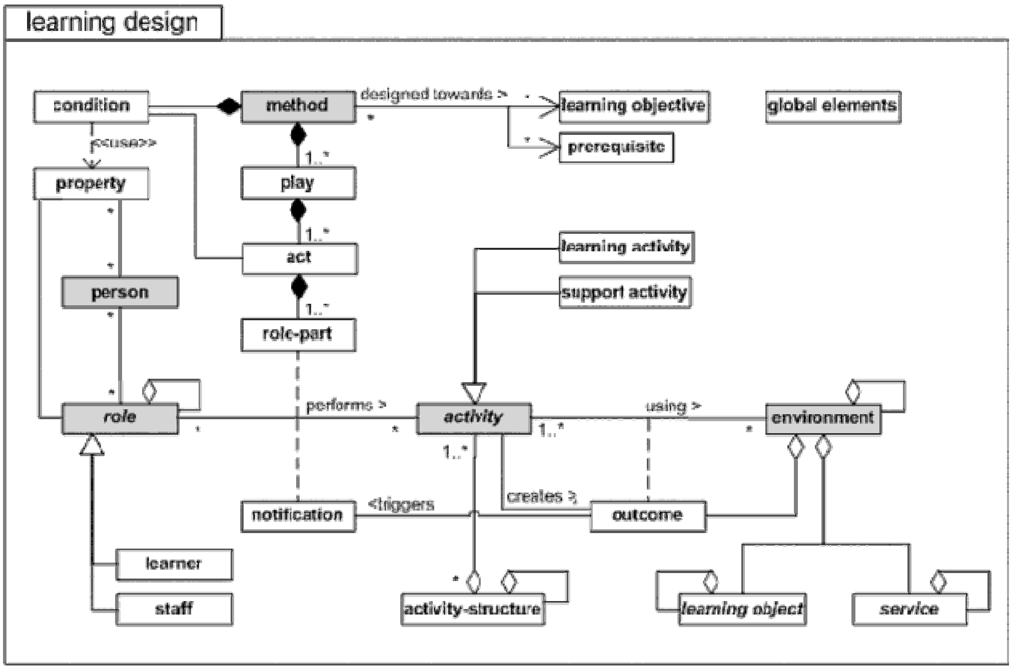


Figure 15. Arquitectura interna y dependencias de IMS-LD

La especificación IMS-LD se ha diseñado con el objetivo de beneficiar, básicamente, a dos grupos: Los profesores y especialistas en educación, y los alumnos

Profesores, diseñadores y proveedores de productos educativos

Una larga tradición nos conduce hasta los modelos educativos actuales, pero su aplicación está limitada por las especificaciones de *eLearning* interoperables [56; 57]. Estas restricciones pueden superarse con el IMS-LD que permite, a los profesores y educadores, ir más allá de la simple construcción de cursos basados bloques de contenidos y dirigidos a un único tipo de alumno. Ahora, el profesor solamente tiene que definir las actividades a realizar por uno o más tipos de alumnos de un entorno compuesto, a su vez, por servicios y recursos. A diferencia de anteriores especificaciones de *eLearning* es posible, además, definir el papel y las actividades del profesor como parte del escenario de aprendizaje en línea.

IMS-LD extiende el concepto de reutilización. En la educación presencial, el profesor se preocupa no sólo por proporcionar recursos pedagógicos para el aprendizaje, sino también en cómo se utilizan estos recursos y en qué contexto. Ahora es posible representar todo esto en un entorno en línea, mediante Unidades de Aprendizaje, con la interoperabilidad garantizada. En este sentido, IMS-LD ofrece una posibilidad nueva: Representar un modelo educativo y hacerlo explícito. Esto abre nuevas perspectivas para futuras discusiones sobre el aprendizaje, basadas en ejemplos claros funcionando en la práctica.

Alumnos

El aprendizaje usando IMS-LD puede ser más interactivo, atractivo y entretenido que con anteriores especificaciones *eLearning* interoperables, ya que ofrece una gama de actividades mayor, e incluso ilimitada.

Con la especificación IMS-LD es posible incluir interacción de apoyo entre los alumnos y con los profesores, de forma que los alumnos pueden aprender unos de otros y los profesores pueden proporcionarles el soporte básico y la realimentación apropiada

Aprender usando IMS-LD puede ser más eficaz porque permite a los diseñadores crear cursos de eLearning basados en cualquier modelo pedagógico empleado en la educación presencial.

Todo ello trae como consecuencia que el esfuerzo invertido por alumno se vea compensado por la obtención de mejores resultados en su aprendizaje [58].

2.6 . Caso de estudio Nivel A: Cándidas

En esta sección describimos una unidad de aprendizaje (UoL) de Nivel A, que muestra la estructura básica de un manifiesto IMS-LD. La utilizaremos a modo de ejemplo para explicar desde la práctica los distintos elementos fundamentales de la especificación codificados. En las UoLs posteriores, que se analizan desde casos de estudio específicos, el grado de detalle será menor, explicando únicamente las partes específicas objeto del escenario y no así las estructuras globales, cosa que hacemos aquí. Todos ellos se incluyen en los Anexos (Sección 10), de una manera detallada, e incorporando las conclusiones y comentarios sobre su diseño y desarrollo a lo largo del cuerpo principal de esta tesis, integradas en los apartados correspondientes. Con esta medida, pretendemos facilitar el seguimiento del hilo de lectura del cuerpo principal.

En concreto, hemos implementado la UoL llamada “Candidas: The Great Unknown” y, que como todas las Unidades de Aprendizaje que hemos desarrollado y que explicamos en esta tesis se encuentran disponibles y operativas como paquetes de información ejecutables en el proyecto Learning Network for Learning Design [59] y en el repositorio Dspace [60] de la Open University of The Netherlands (Figure 16).

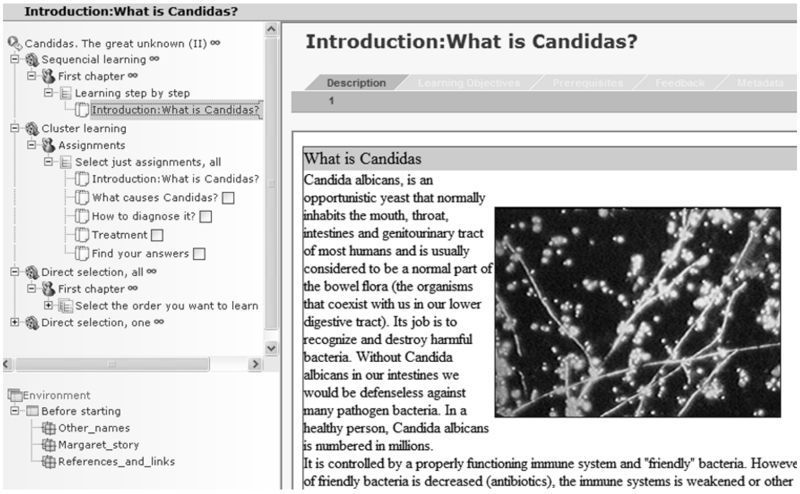


Figure 16. Caso de estudio Nivel A: Cándidas

Esta UoL formaliza un curso sobre un tema concreto, en este caso un tipo de hongo intestinal, de manera lineal. Solamente existe un rol de estudiante, un play, un acto y una estructura de aprendizaje compuesta por varias actividades de aprendizaje y entornos con información complemetaria asociados. Existen a su vez diversos recursos externos enlazados como tipo `webcontent` a las actividades de aprendizaje y a los entornos.

El diagrama UML de la unidad se muestra en la siguiente ilustración (Figure 17):

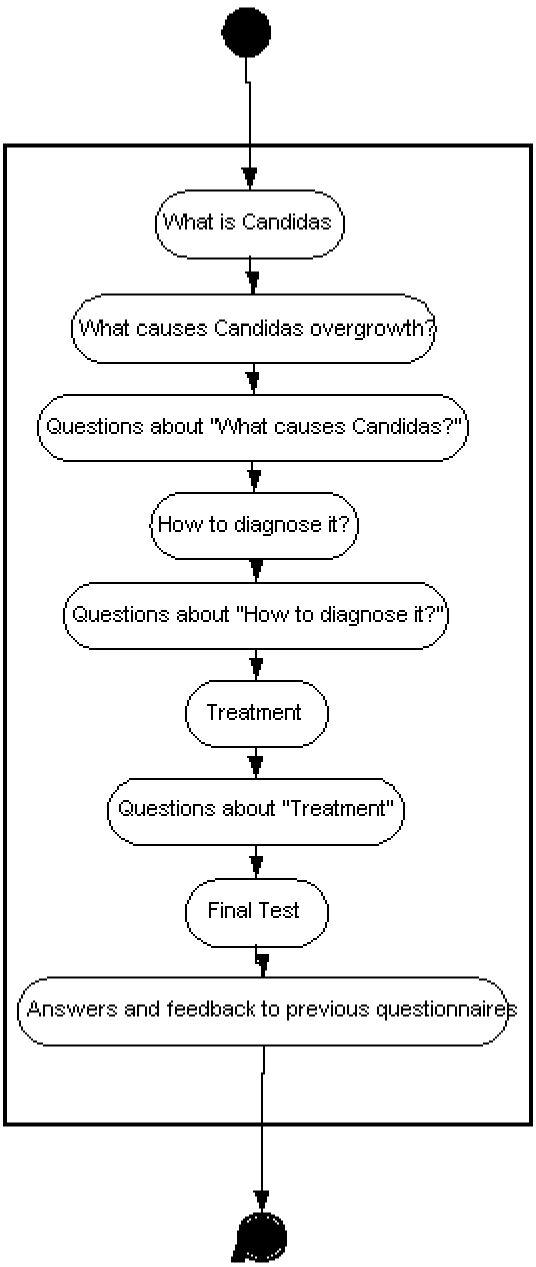


Figure 17. Diagrama UML de la UoL Candidas

El manifiesto define el diseño instructivo de IMS-LD compuesto por una serie de componentes[1]:

Referencia al esquema XML

```
<?xml version="1.0"?>
<!-- Candidas. The great unknown (I), r2.0, 10 Nov 2004 -->
<!-- schemas -->
```

[1] Todo el código aportado de ahora en adelante refleja piezas seleccionadas de un fichero consistente, y a veces adaptadas a pseudocódigo, con objeto de lograr una mayor inteligibilidad, no suponiendo nunca extractos ejecutables directamente si no es mediante la debida inclusión en un formato completo válido de una unidad de aprendizaje

```
<manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1" xmlns:IMS-
LD="http://www.imsglobal.org/xsd/IMS-LD_v1p0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1
http://www.imsglobal.org/xsd/imscp_v1p1p3.xsd
http://www.imsglobal.org/xsd/IMS-LD_v1p0
http://www.imsglobal.org/xsd/IMS_LD_Level_A.xsd" identifier="example-
candidas">
      <metadata>
            <schema>IMS Metadata</schema>
            <schemaversion>1.2</schemaversion>
      </metadata>
      <organizations>
            <learning-design xmlns="http://www.imsglobal.org/xsd/IMS-
LD_v1p0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/IMS-LD_v1p0
http://www.imsglobal.org/learningdesign/ldv1p0/IMS_LD_Level_A.xsd"
identifier="Course-candidas" level="A" uri="http://ou.nl/examplecandidas">
```

Nota: El Nivel de IMS-LD se especifica así como un identificador único URI, además de los esquemas XML y XSD de la especificación y del XML que lo soporta. El learning-design se encuentra dentro de la sección organizations del paquete de información

Título

```
<title>Candidas. The great unknown (I)</title>
```

Nota: Todo lo que viene a continuación hasta el método se acota entre etiquetas <components>

Roles

```
<roles>
      <learner identifier="Learner">
            <title>Learner</IMS-LD:title>
      </learner>
</roles>
```

Nota: En este caso existe un único rol de tipo Learner. Puede haber tantos roles tipo Learner como se desee (ej. Distintos grupos de estudiantes) y puede haber además uno o varios roles de tipo Staff (ej. Profesor, Administrador, Profesor-Prácticas)

Actividades de aprendizaje

```
<learning-activity identifier="Preparation">
```

```
        <title>Introduction:What is Candidas?</title>
        <environment-ref ref="resources-Preparation" />
        <activity-description>
              <item identifierref="R-Preparation" identifier="I-
preparation"/>
        </activity-description>
        <complete-activity>
              <user-choice/>
        </complete-activity>
</learning-activity>
```

Nota: Se definen un total de 9 actividades de aprendizaje entre etiquetas `<activities>`. En caso de actividades de soporte, el sistema de definición es idéntico, pero con una etiqueta distinta. Dentro de la actividad se define un título y una `Activity-Description` de manera obligatoria. También se pueden indicar los entornos asociados y la forma de terminación (ej. Elección de usuario, valor de una variable)

Estructura de actividades

```
<activity-structure identifier="AS-learningactivity" structure-
type="sequence">
        <title>Activities for chapter 1</title>
        <learning-activity-ref ref="Preparation"/>
        <learning-activity-ref ref="Assignment-1"/>
        <learning-activity-ref ref="Test-1"/>
        <learning-activity-ref ref="Assignment-2"/>
        <learning-activity-ref ref="Test-2"/>
        <learning-activity-ref ref="Assignment-3"/>
        <learning-activity-ref ref="Test-3"/>
        <learning-activity-ref ref="Test-final"/>
        <learning-activity-ref ref="Feedback"/>
</activity-structure>
```

Nota: Se indica la forma de visualización de las actividades dentro de la estructura (ej. `Sequence` o `Selection` indicando el número mínimo de actividades a completar)

Entornos

```
<environment identifier="resources-Preparation">
        <title>Before starting</title>
        <learning-object identifier="Other_names">
              <item identifierref="R-before1" identifier="I-before1"/>
        </learning-object>
        <learning-object identifier="Margaret_story">
```

```
            <item identifierref="R-before2" identifier="I-before2"/>
      </learning-object>
      <learning-object identifier="References_and_links">
            <item identifierref="R-before3" identifier="I-before3"/>
      </learning-object>
</environment>
```

Nota: Entre etiquetas <environments> se definen los entornos necesarios asociados a actividades y actos. Dentro de cada entorno puede haber referencia a diversos recursos y servicios

Método

```
<method>
      <play>
            <title>Basic Example Level A</title>
            <act>
                  <title>First chapter</title>
                  <role-part>
                        <title>Role part learner</title>
                        <role-ref ref="Learner"/>
                        <activity-structure-ref ref="AS-learningactivity"/>
                  </role-part>
            </act>
      </play>
</method>
```

Nota: Un role-part por cada actividad o grupo de actividades por rol

Recursos

```
<resource identifier="R-before1" type="webcontent" href="res_before1.html">
      <file href="res_before1.html"/>
</resource>
```

Nota: Tantos recursos (en este caso, de tipo webcontent con un fichero formato HTML) como estén referenciados en las secciones Activity-Description y Environment, entre etiquetas <resources>

A partir de esta estructura básica se pueden incorporar y modificar diversos componentes, como la definición de objetivos y de pre-requisitos, la ejecución de estructuras de actividad de manera indexada (selection) o la inclusión de un servicio de comunicación tipo foro. También se pueden añadir elementos avanzados, tal y como describimos en la siguiente sección dedicada a los niveles B y C.

2.7. IMS-LD: Level B y Level C

La mayoría de las herramientas y de los cursos de ejemplo existentes se basan en el Nivel A de la especificación. Esto es debido a varios factores: a) Es el nivel más sencillo de implementar tanto para diseñadores de aprendizaje como para desarrolladores de software [61]; b) el método pedagógico mayoritario que utilizan los profesores se ajusta perfectamente al Nivel A [62]; y c) no existen en la actualidad herramientas que permitan un diseño de alto nivel que abarque los elementos de los Niveles B y C, por lo que los creadores no técnicos no pueden hacer uso de él [53]. Sin embargo, los Niveles B y C proporcionan unos elementos que dotan a los diseños de aprendizaje de más versatilidad y capacidad, como veremos en la sección siguiente. Estos elementos son a) `properties`, b) `global elements`, c) `monitor services`, d) `calculations`, e) `conditions` y f) `notifications`.

a) `Properties`: Variables para almacenar información del sistema, de los usuarios y de los grupos de usuarios. Pueden ser de varios tipos, como de tipo texto, booleano, entero o fichero. Las propiedades deben ser declaradas y pueden ser inicializadas en el manifiesto de IMS-LD. Posteriormente son utilizadas en el propio manifiesto (por ejemplo, como condición de terminación de una actividad, o como operando de una condición o de una operación) o en ficheros externos XHTML (por ejemplo, visualizando o introduciendo el valor asociado. Los ficheros XHTML son declarados en el manifiesto como de tipo `imsldcontent` (mientas que un recurso no XML o XHTML es de tipo webcontent) y se incorporan al paquete de información, es decir, a la unidad de aprendizaje. Para que los ficheros XHTML pueden ver e introducir los valores de las variables hay que utilizar global elements.

Hay cinco tipos de propiedades, que pueden ser agrupadas:

- `Local properties`: Únicamente accesible dentro del contento de una instancia de una unidad de aprendizaje. Es decir, la UoL se publica y luego se instancia tantas veces como se requiera ejecutar. La `local property` únicamente será accesible para cada instancia de manera independiente. Así, cada vez que se ejecuta una nueva instancia, la propiedad puede tener un nuevo valor. A su vez, dentro de la instancia, la variable tiene el mismo valor para todos los roles y usuarios. Por ejemplo, un mensaje de bienvenida al curso

```
<loc-property identifier="Welcome">
    <title>Welcome message</title>
    <datatype datatype="text" />
    <initial-value>Nice to see you again</initial-value>
</loc-property>
```

- `Local personal properties`: También restringida a cada instancia, pero con un valor distinto para cada usuario. Por ejemplo, la calificación de un ejercicio

```
<locpers-property identifier="Grade">
    <title>Grade</title>
    <datatype datatype="string" />
</locpers-property>
```

- `Local role properties`: También restringida a cada instancia, con un valor distinto para cada rol. Por ejemplo, existen dos roles de tipo `Learner` (grupo 1 y grupo 2) y cada uno debe proporcionar una respuesta grupal a un problema dado común

```
<locrole-property identifier="Pregunta4">
 <title>Pregunta tipo combobox alfanumérico</title>
 <datatype datatype="string" />
 <initial-value>Select</initial-value>
  <restriction restriction-type="enumeration">Select</restriction>
  <restriction restriction-type="enumeration">Lunes</restriction>
  <restriction restriction-type="enumeration">Martes</restriction>
  <restriction restriction-type="enumeration">Miércoles</restriction>
</loc-property>
```

- `Global personal properties`: Disponible en cada instancia con el mismo contenido para un usuario, pero distintos contenidos para distintos usuarios. Por ejemplo, el campo email de la ficha personal de cada estudiante

```
<globpers-property identifier="prop-1">
      <global-definition uri="http://whoknows.nl">
            <title>Email address</title>
            <datatype datatype="string" />
            <initial-value>daniel.burgos@ou.nl</initial-value>
      </global-definition>
</globpers-property>
```

- `Global properties`: A modo de variable global, se puede definir y utilizar en todas las instancias por todos los usuarios de todos los roles. Por ejemplo, un mensaje de alerta sobre el uso del sistema

```
<glob-property identifier="Alert">
      <global-definition uri="http://whoknows.nl">
            <title>Email2</title>
            <datatype datatype="string" />
            <initial-value>Mensaje de alerta general</initial-value>
      </global-definition>
```

```
</glob-property>
```

b) `Global elements`: Son elementos IMS-LD que pueden ser incrustados en ficheros XML o XHTML externos utilizando el mismo namespace. Estos ficheros externos deben ser de tipo `imsldcontent`. El resultado final es que contenidos HTML, XML y IMS-LD pueden ser integrados. En la práctica, un global element puede ver y configurar una propiedad o grupo de propiedades ya definidas en el manifiesto:

- `view-property` and `view-property-group`: Muestra el contenido de una propiedad o de un grupo de propiedades

- `set-property` and `set-property-group`: Solicita la introducción del valor de una propiedad o grupo de propiedades. Tiene un atributo opcional `max-transactions` para establecer el número de veces máximo que un usuario puede introducir un valor

Cada tipo de propiedad se presenta de una manera distinta, dependiendo de si se visualiza/solicita un tipo texto, numérico o combobox, por ejemplo.

```
<set-property ref="accuracy" property-of="self"/>
```

También un global elemente puede mostrar y ocultar capas de tipo DIV definidas en un fichero DHTML. Así, el contenido del manifiesto sería:

```
<show>
      <class class=" Answer1_Right" />
</show>
```

Mientras que el contenido del fichero DHTML (extensión .XML) asociado sería:

```
<div class="Answer1_Right">
      <font color="black">Congratulations! It's the right answer</font>
      <img src="eiffel.jpg"></img>
</div>
```

c) `Monitor services`: Las propiedades ajenas pueden ser vistas y configuradas mediante el servicio de monitorización. Dentro del entorno de una actividad se define el elemento monitor y asociado a un global element se define a quién pertenece la propiedad correspondiente, si a uno mismo (self) o a otros usuarios del mismo rol (supported-person)

```
<view-property ref="accuracy" property-of="supported-person" view="value"/>
```

d) `Calculations`: Dentro del manifiesto, se pueden realizar operaciones aritméticas sencillas (adición, sustracción, división y multiplicación) con distintas variables, así como con valores constantes, y asignaciones de los resultados a otras variables. Las operaciones pueden estar anidadas. Los tipos de los operandos deben coincidir, aunque no se realiza ninguna comprobación hasta el momento de validación previo a la publicación de la UoL. El ejemplo siguiente opera la siguiente fórmula: `Total=Value1+Value2+Value3+Value4+Value5`.

```
<change-property-value>
    <property-ref ref="total" />
        <property-value>
        <calculate>
        <sum>
        <property-ref ref="Value5" />
            <sum>
            <property-ref ref="Value4" />
                <sum>
                <property-ref ref="Value3" />
                    <sum>
                          <property-ref ref="Value1" />
                          <property-ref ref="Value2" />
                    </sum>
                </sum>
            </sum>
        </sum>
        </calculate>
    </property-value>
</change-property-value>
```

e) `Conditions`: Dentro del manifiesto, IMS-LD admite el formato de condición simple `if-then-else`, permitiendo también la anidación de cuantas condiciones simples se requiera. No hay estructuraras condicionales complejas (tipo `case`). La comparación de la condición contempla los operadores relacionales habituales, tanto para variables y valores numéricos como alfanuméricos. Esta estructura condicional constituye la forma básica de definición y gestión de comportamientos dinámicos (ej. Aprendizaje adaptativo basado en el rendimiento del usuario, personalización basada en la decisión personal del profesor). En el ejemplo siguiente, si el valor de la propiedad `Answer1` es `5`, entonces el valor de la propiedad `points-right` se asigna a la propiedad `Value1`, en caso contrario, `Value1` contendrá el valor numérico real `0.0`.

```
<if>
    <is>
        <property-ref ref="Answer1" />
        <property-ref ref=5/>
    </is>
```

```
</if>
    <then>
        <change-property-value>
              <property-ref ref="Value1" />
              <property-value>
                <property-ref ref="points-right" />
              </property-value>
        </change-property-value>
    </then>
          <else>
            <change-property-value>
              <property-ref ref="Value1" />
              <property-value>0.0</property-value>
            </change-property-value>
          </else>
```

f) Notifications: El Nivel C está construido únicamente alrededor de las notificaciones, que permiten lanzar automáticamente un mensaje de correo electrónico o activar un flag de una actividad de aprendizaje. Para ello, debe existir previamente una variable global con la dirección del destinatario y se debe hacer definido el protocolo SMTP de correo saliente en la configuración del motor IMS-LD.

```
<notification>
    <email-data email-property-ref="prop-1">
        <role-ref ref="role-student" />
    </email-data>
    <learning-activity-ref ref="la-1" />
    <subject>Mensaje de notificacion</subject>
</notification>
```

2.8 . Principales recursos didácticos y utilización del Nivel B y del Nivel C

Todas las facilidades que proporcionan el Nivel B y el Nivel C en cuanto a propiedades, condiciones, servicio de monitorización y demás elementos, pueden ser utilizadas en una amplia gama de aplicaciones. A continuación describimos las más frecuentes desde un punto de vista didáctico: active learning, collaborative learning, adaptive learning and personalisation, dynamic feedback, runtime tracking, ePortfolios y new ways of assessment[2].

2.8.1. Active and collaborative learning

El aprendizaje colaborativo consiste [63] en la compartición de información alumno-alumno y alumno-profesor con el fin de afianzar conocimientos, criticar puntos de vista, aportar nuevas ideas a la luz del trabajo de otros o introducir conceptos para discusión y elaboración grupal, por ejemplo. Esto implica que la misma información puede ser vista por distintas personas y que se produce un flujo constante de intercambio de datos que cada cual utilizará según sus objetivos personales dentro del grupo y del curso.

Una segunda posibilidad es la monitorización del progreso del estudiante por parte del tutor, analizando dinámicamente las aportaciones que este vaya realizando durante el curso y proporcionando un feedback al mismo. De esta manera, se establece también un flujo bilateral de información entre alumno y tutor orientado al aprovechamiento académico y personalizado.

El Nivel B proporciona el objeto 'monitor' que permite visualizar propiedades propias o ajenas de una manera estructurada. Estas propiedades deben ser definidas con anterioridad, e inicializadas para tipos numéricos, y pueden ser operadas mediante, como veremos más adelante. Fijémonos en dos ejemplos sobre definición e inicialización:

```
<loc-property identifier="LP-LA-1-completed">
  <title>Respuesta al cuestionario inicial</title>
  <datatype="boolean"/>
  <initial-value>true</initial-value>
</loc-property>
```

La propiedad 'LP-LA-1-completed', cuyo título o etiqueta es 'Respuesta al cuestionario inicial', es de tipo lógico y su valor inicial es TRUE. El uso de esta variable será almacenar el estado terminado o no de una actividad de aprendizaje.

```
<locpers-property identifier="LP-personalgoals">
  <title>Qué objetivos tienes para el curso?</title>
  <datatype="text"/>
</locpers-property>
```

La propiedad 'LP-personalgoals', cuyo título o etiqueta es 'Qué objetivos tienes para este curso?', es de tipo texto y no se inicializa con ningún valor. El objetivo de esta

[2] Como ya se indicó previamente, el código aportado ha sido seleccionado y/o modificado con pseudocódigo, que mantiene intacta la estructura, para obtener una mayor legibilidad

variable será que el usuario pueda introducir comentarios y, por tanto, se inicializa en blanco. Es una propiedad local y depende de cada usuario.

Tomando esta última variable ‘LP-personalgoals’ puede ser leída y grabada dentro del flujo normal del curso:

```
<set-property ref="LP-personalgoals"/>
```

a) permite escribir (‘set’) el contenido de la variable.

```
<view-property ref="LP-personalgoals" property-of=”self/>
```

b) permite leer (‘view’) el valor (‘value’) de nuestra propia variable (‘self’).

Pero también puede ser monitorizada mediante el objeto ‘monitor’ que puede recoger esta variable para realizar su seguimiento. Por ejemplo, la siguiente línea nos permite leer (‘view’) la variable de otro estudiante (‘supported-person’):

```
<view-property property-of="supported-person" ref="LP-personalgoals"/>
```

Para realizar esta acción, primero deberemos configurar el objeto ‘monitor’ dentro del entorno de trabajo (en este caso concreto):

```
<environment identifier="E-personalgoals">
  <title>Qué objetivos tienen los demás?</title>
  <service identifier="S-personalgoals">
    <monitor>
      <role-ref ref="Student"/>
      <title>Metas de los demás alumnos</title>
      <item identifierref="R-personalgoals"/>
    </monitor>
  </service>
</environment>
```

Como podemos ver, se define el servicio de monitor para el estudiante (‘Student’). Esto permite que cada estudiante pueda visualizar el contenido de las variables de sus compañeros. Para la visualización por parte del tutor se construiría una estructura similar, facilitando un seguimiento de los participantes en el curso. La gestion y visualización del conjunto de variables proporciona una mayor trasparencia al proceso del aprendizaje.

2.8.2. Adaptive learning and personalisation

Para ilustrar esta sección nos basaremos en el ejemplo ‘Learning to listen to Jazz’ [59] desarrollado originalmente en EML y adaptado a IMS-LD. En él un alumno/a puede seguir un curso y elegir dos itinerarios distintos, temático e histórico en función de sus preferencias. En el ejemplo también se realizan funciones de monitorización como las descritas en el apartado anterior.

El aprendizaje adaptativo busca, partiendo de una base de contenidos comunes, elegirlos y presentarlos al estudiante, en función de su perfil inicial y de los resultados progresivos que vaya proporcionando durante el desarrollo del curso [64]. La personalización complementa el aprendizaje adaptativo proveyendo a cada estudiante la capacidad de elegir ciertas características del contenido, de la presentación, de la evaluación y del itinerario, dentro de un marco prefijado de aprendizaje.

En referencia al aprendizaje adaptativo, en el ejemplo referenciado, Jazz, se define la propiedad 'LP-choose-itinerary' para saber si el usuario ha elegido o no el itinerario. Cada itinerario se encuentra descrito en una estructura de actividad, 'AS-historic' y 'AS-thematic', definidas previamente y que no detallamos aquí. Todo el proceso se detalle como un conjunto de condiciones. En caso de que todavía no se haya elegido nada, y la variable no tenga ningún valor aún, se esconden las dos Activity Structures:

```
<conditions>
<if>
  <no-value>
    <property-ref ref="LP-choose-itinerary"/>
  </no-value>
</if>
<then>
  <hide>
    <activity-structure-ref ref="AS-thematic"/>
    <activity-structure-ref ref="AS-historic"/>
  </hide>
</then>
```

El estudiante puede elegir las dos opciones 'historic' y 'thematic' presentes en una combobox. Si el usuario elige la opción 'thematic', la estructura 'AS-thematic' se muestra y la 'AS-historic' se oculta (Figure 18). Lo mismo en el caso contrario:

```
<if>
  <is>
    <property-ref ref="LP-choose-itinerary"/>
    <property-value>thematic</property-value>
  </is>
</if>
<then>
```

```
  <show>
    <activity-structure-ref ref="AS-thematic"/>
  </show>
  <hide>
    <activity-structure-ref ref="AS-historic"/>
  </hide>
</then>
```

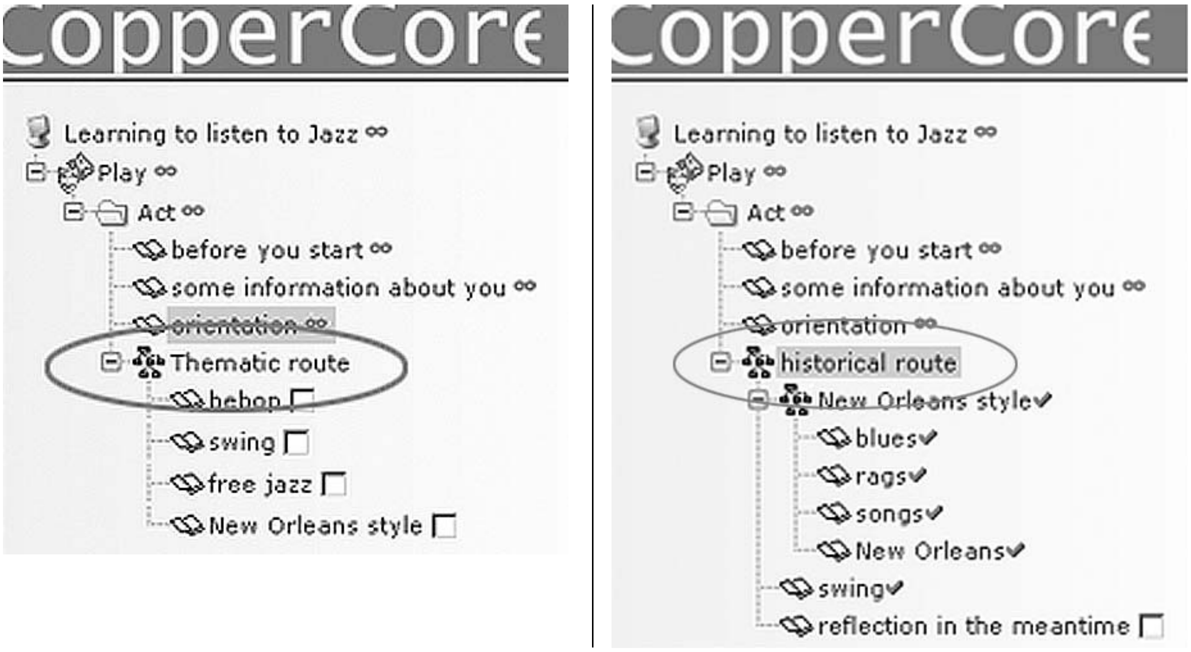


Figure 18. Itinerarios Historic y Thematic en el example Jazz

Las diferentes estructuras pueden contener distintos contenidos o los mismos contenidos reorganizados de diferentes maneras para dos enfoques distintos y todo estar gestionado dentro del propio manifiesto de la Unidad de Aprendizaje.

Por otro lado, un caso sencillo de personalización es la incorporación de los datos personales del alumno/a en el desarrollo del curso. Ya hemos visto en el apartado anterior cómo definir y rellenar una propiedad. Otra posibilidad consiste en utilizar un grupo de propiedades para tener una operativa más sencilla. Por ejemplo:

a) definiendo una propiedad de tipo ‘string’ y una propiedad de tipo ‘integer’, e inicializando esta última a cero

```
<locpers-property identifier="LP-name">
  <title>your name</title>
  <datatype datatype="string"/>
  </locpers-property>
<locpers-property identifier="LP-age">
  <title>age</title>
  <datatype datatype="integer"/>
```

```
  <initial-value>0</initial-value>
</locpers-property>
```

b) agrupando las dos propiedades

```
<property-group identifier="LP-group-personalinfo">
  <title>personal details</title>
  <property-ref ref="LP-name"/>
  <property-ref ref="LP-age"/>
</property-group>
```

c) solicitando al usuario los datos (`‘set’`) y visualizándolos (`‘view’`), según necesidad

```
<set-property-group ref="LP-group-personalinfo" property-of="self"/>

<view-property-group ref="LP-personalgoals" property-of="self"/>
```

2.8.3. Dynamic feedback and runtime tracking

Para ilustrar esta sección utilizaremos el ejemplo ‘GeoQuiz’ [59] en el que el alumno/a contesta un cuestionario de cinco preguntas que devuelve información adicional según las respuestas elegidas y que proporciona una valoración númerica final del ejercicio de evaluación del estudiante.

Como aplicación concreta del aprendizaje adaptativo, IMS-LD también permite trabajar con retroalimentación dinámica, proporcionando contenidos adaptados a la evolución del estudiante dentro de cada `Learning Activity` y lectura de resultados en función de la actividad del mismo.

En referencia a la aportación de contenidos en función de la evolución ya hemos visto en el ejemplo anterior que una posibilidad dentro del propio manifiesto es la utilización de diferentes estructuras con contenidos iguales o distintos y distinta reorganización dentro del manifiesto. En definitiva, es ocultar o mostrar una actividad o una estructura, que al final estarán enlazadas con recursos apuntando a ficheros externos. Otra posibilidad consiste en la utilización de clases para modificar la visualización del propio contenido dentro de un fichero apuntado por un recurso, es decir, fuera del manifiesto. Estas clases hacen referencia a capas DHTML definidas en ficheros XML aparte de imsmanifest.xml.

Aunque el mecanismo que se puede utilizar para visualizar y ocultar las capas es similar al visto en el apartado anterior para `Activity Structures`, la acción apuntada se realiza en otro fichero y se ejecuta, por tanto, de otra manera:

```
<if>
  <is>
    <property-ref ref="Answer1"/>
```

```
    <property-value>C</property-value>
  </is>
</if>
<then>
  <hide>
    <class="Feedback_Wrong"/>
  </hide>
  <show>
    <class="Feedback _Right"/>
  </show>
</then>
```

En el ejemplo anterior se muestra la clase ‘Feedback_Right’ si el contenido de la respuesta ‘Answer1’ es ‘C’ (que se elige de una lista enumerada), ocultando la clase ‘Feedback_Wrong’, que contiene otro contenido distinto. En el fichero externo, se establece el valor de la variable de respuesta ‘Answer1’ y se definen tanto las clases como el contenido de las mismas:

a) se elige la respuesta (‘set’)

```
<p>Your answer is:
<set-property ref="Answer1" of="self"/></p>
```

b) se definen las clases

```
<div class="Answer1_Wrong">
  <p><view-property ref="Answer1"/> is not right</p>
</div>
<div class="Answer1_Right">
  <p>Congratulations!</p>
  <img src="eiffel.jpg"/>
</div>
```

En referencia a la lectura de resultados en función de la actividad del usuario, podemos realizar operaciones matemáticas con los valores almacenados en el transcurso del estudio y proveer un feedback contextual de los mismos. Si imaginamos dos preguntas con dos respuestas y dos valores posibles para cada respuesta, 0 y 100, podríamos:

a) definir cada propiedad

```
<locpers-property identifier="QuestionTrue1">
  <datatype datatype="integer"/>
  <initial-value>0</initial-value>
</locpers-property>
```

b) asignarle un valor en función de la respuesta

```
<if>
  <is>
    <property-ref ref="Answer1"/>
    <property-value>C</property-value>
  </is>
</if>
<then>
  <change-property-value>
    <property-ref ref="QuestionTrue1"/>
    <property-value>100</property-value>
  </change-property-value>
</then>
<else>
  <change-property-value>
    <property-ref ref="QuestionTrue1"/>
    <property-value>0</property-value>
  </change-property-value>
</else>
```

c) y, finalmente, realizar la media aritmética de las dos respuestas

```
<change-property-value>
  <property-ref ref="sum"/>
  <property-value>
    <calculate>
      <divide>
        <sum>
          <property-ref ref="QuestionTrue1"/>
          <property-ref ref="QuestionTrue2"/>
        </sum>
        <property-value>2</property-value>
      </divide>
    </calculate>
  </property-value>
</change-property-value>
```

2.8.4. **ePortfolios y nuevas formas de evaluación**

La evaluación tradicional se basa en la confrontación de los conocimientos de un usuario frente a la máquina o frente a los conocimientos del profesor. Pero no es la única. Las nuevas formas de aprender y de enseñar deben ir de la mano de nuevas formas de evaluar. La evaluación debe concebirse dentro del flujo de aprendizaje y no como un recurso aislado a) comprobando el nivel de conocimiento de un estudiante antes de decidir el mejor itinerario, b) comprobando si un concepto ha quedado fijado antes de avanzar hacia el siguiente y c) proporcionando feedback de alta calidad para mantener una motivación alta. También deben contemplarse d) los escenarios compuestos por varios estudiantes compartiendo, por ejemplo, respuestas individuales con el grupo para una evaluación y un debate colectivos [52]. Formalizar cualquiera de estas propuestas en IMS-LD no es difícil, aunque sí laborioso, si utilizamos los apuntes descritos hasta ahora. Las cuatro posibilidades pueden ser implementadas como combinación de condiciones y propiedades.

Del mismo modo, como ya vimos en el apartado anterior, la formalización de cuestionarios de evaluación es posible dentro de IMS-LD, como también lo será la integración de esta especificación con Question and Test Interoperability [21]. Si bien la codificación de QTI no es objeto de esta sección, sí podemos apuntar cómo es la actual integración de QTI en IMS-LD. Básicamente, la secuencia es la misma que la comentada anteriormente para describir la relación entre un entorno y un recurso asociado. La diferencia está en la referencia del recurso tipo QTI (‘imsqti_item_xmlv2p0’) que apuntaría a un fichero escrito en en esta especificación (‘question_1.xml’):

```
<environment identifier="Env-1">
       <title>A test linking QTI and IMS-LD</title>
       <learning-object identifier="LO-QTI-question-1">
            <title>First question</title>
      <item identifier="I-1" identifierref="R-que-1"/>
      </learning-object>
</environment>
```

```
<resource identifier="R-que-1" type="imsqti">
      <file href="question_1.xml"/>
</resource>
```

Por último, la utilización de ePortfolios es posible si mantenemos el espíritu propio de IMS ePortfolio [65], es decir, que la grabación y compartición de datos externamente a cada aplicación se realice de manera estandarizada dentro del paquete de información, posibilitando que la información pueda ser utilizada en sitios diferentes por diferentes sistemas. Las propiedades globales que tiene IMS-LD permiten definir información de un usuario (globpers-property), de un grupo (property-group) o, a modo de constantes, para todo el mundo (glob-property). La primera es llamada también portfolio-property porque permite la incorporación de ficheros a la ejecución de la unidad de aprendizaje. Como siempre, la utilización requiere dos momentos distintos:

a) la creación e inicialización de la variable global

```
<globpers-property identifier="GP-suggestions">
  <global-definition uri="GP-suggestions ">
    <title>Suggestions about the course</title>
    <datatype datatype="file"/>
  </global-definition>
</globpers-property>
```

b) la utilización de la propiedad en un fichero XML

```
<div class="upload-file">
    <p>Choose the file with your suggestions:</p>
    <set-property ref="GP-suggestions"/>
</div>
```

2.9 . **Reflexiones sobre los Niveles B y C**

Aunque el Nivel A proporciona el esqueleto y la base de una unidad de aprendizaje con su itinerario formativo, el Nivel C y sobre todo el Nivel B proveen de los recursos más potentes y versátiles. Mientras que el Nivel A define el 80% de una UoL a nivel estructural, los Niveles B y C definen el 80% de la potencialidad. Con la adecuada combinación de los elementos descritos se pueden abordar retos actuales de la enseñanza como los enunciados (active learning, collaborative learning, adaptive learning, runtimet tracking, etc). Aun así, hay dos grandes obstáculos aún: El primero se centra en las estructuras tan primarias y escasas de programación. Sin estructuras condicionales complejas, bucles, grabación/recuperación de datos en disco o arrays, por ejemplo, las posibilidades de flexibilidad de la especificación son limitadas. El segundo obstáculo radica en la propia sintaxis y gramática basada en lenguajes de marcas tipo XML. Es una notación compleja para realizar operaciones sencillas, lo que eleva el nivel técnico que una persona debe poseer para poder aplicarla. Con los editores actuales, que ya hemos indicado se encuentran muy centrados en la solución técnica y no en la metáfora o utilización pedagógica en sí misma, la edición de un Nivel A se hace llevadera. Pero la edición de un Nivel B/C es extremadamente laboriosa y encriptada, resultando más fácil abordar la creación de este tipo de UoLs directamente con un editor plano de propósito general (tipo bloc de notas) o con un editor tipo XML. Esta edición más técnica, vuelve a restringir las posibilidades de utilización directa por parte de la comunidad virtual de usuarios no técnicos (profesores, proveedores de contenidos, estudiantes no tecnológicos...) y también dificulta una implementación sencilla y ágil incluso por parte de usuarios técnicos experimentados (diseñadores de aprendizaje, desarrolladores). La edición debería estar lo más independiente posible de la estructura técnica subyacente, incluso resultando opaca la división por niveles o la sintaxis. Un enfoque más visual, con editores de alto nivel, y una metáfora de utilización más próxima a la realidad cotidiana de profesores y alumnos no técnicos, supone un avance necesario para el uso correcto, extendido y real de la especificación con todo su potencial. Esto no quita que sigan existiendo los editores técnicos y de propósito general para aquellos conocedores de la notación y de IMS-LD en profundidad, editores de los que ya disponemos y que ya han sido mencionados (CopperAuthor, Reload LD Editor, Cosmos…).

2.10 . **Crítica a IMS-LD desde los casos de estudio**

Con el propósito de realizar un análisis preciso sobre la especificación, hemos diseñado y modelado un conjunto de casos de estudio que pueden encontrarse en los anexos (Sección 10), y que afectan a la estructura general de IMS-LD. En concreto, referentes a los niveles B y C, son *Caso de estudio Nivel B: Global Warming* y *Caso de estudio Nivel C: Ampliación del Nivel B*. Estos ejemplos concretos son bastante representativos de la capacidad de cada nivel. Cada ejemplo aporta nuestras reflexiones y comentarios sobre la expresividad pedagógica de IMS-LD. Nos muestran UoLs reales, con soluciones concretas de codificación a un flujo educativo dado abarcando varios roles, diversos usuarios y metodologías. A continuación, realizamos un análisis y una crítica obtenidos del estudio teórico de la especificación y de la implementación práctica de los ejemplos modelo allí detallados (para Niveles A, B y C) y otros parecidos [58; 61; 66-68], todos disponibles en LN4LD [6] y Dspace [7].

2.10.1. **Desde la notación**

- IMS-LD maneja una gramática incómoda de modelado. Aunque la metáfora que permite definir y relacionar los distintos componentes es clara desde un punto de vista conceptual, no lo es tanto cuando se intenta implementar un ejemplo concreto. La división entre `learning design`, `plays`, `acts` y `activity structures`, por ejemplo, proporciona una relación en cascada que, siendo versátil para crear escenarios complejos, dista mucho de los escenarios sencillos sujetos habitualmente a cursos específicos. Esta subordinación obligatoria de elementos añade confusión. Otro ejemplo muestra las referencias consecutivas que definen un servicio y que necesitan definición en la actividad, el entorno, el servicio en sí, el recurso y el fichero que lo ejecuta
- La notación IMS-LD es difícil de concretar ya que sigue los parámetros habituales XML, algo no trivial incluso para programadores, pero define una estructura propia de

manifiesto Content Packaging con una extensión de la etiqueta organizations, que genera un manifiesto Learning Design extenso y farragoso de seguir. Además, las etiquetas y definición de elementos es muy detallada lo que hace laborioso realizar labores sencillas, como una simple definición de variables, por ejemplo. La programación, por tanto, se vuelve complicada, a pesar de su sencillez

- Las estructuras de programación proporcionadas son muy básicas (condición simple, aritmética simple, edición y visualización de variables, gestión de visibilidad de actividades y capas DIV), además de utilizar una sintaxis extensa, lo que dificulta la visibilidad. Una extensión de estas estructuras y una modificación de su sintaxis y/o gramática aumentaría exponencialmente la potencialidad de la especificación en la flexibilidad pedagógica que busca

2.10.2. **Desde la estructura conceptual y la arquitectura de IMS-LD**

- Aunque hoy en día únicamente la exportación a otras especificaciones parece posible, incluso para el editor más avanzado (fijándonos por ejemplo en Reload LD Editor [5; 69] o en Cosmos [45; 70]) la importación y exportación al 100% representan todavía un obstáculo y constituye por tanto un gran paso para lograr el intercambio efectivo de paquetes de información. De esta manera, se logrará uno de los objetivos de la estandarización del eLearning, como es la interoperabilidad de las unidades o actividades de aprendizaje en diversos sistemas de edición y ejecución, con el objeto de lograr flexibilidad y un mayor grado de autonomía. Esta interoperabilidad es necesaria dentro y fuera de la especificación
- Para lograr la interoperabilidad con otras especificaciones y módulos externos a IMS-LD es necesario un sistema o capa de comunicación que permita el intercambio efectivo de información, como variables y valores de variables y estado, lo que permitirá interactuar desde IMS-LD en el módulo externo y viceversa
- Del mismo modo, junto con la interoperabilidad, el intercambio de paquetes logrará la re-utilización de Unidades de Aprendizaje previamente modeladas, bien para su re-definición, bien para su compilación e inclusión en unidades mayores. Este intercambio de paquetes puede servir de base para el desarrollo de plantillas o patrones de diseños instructivos interoperables, lo que facilitaría la creación y utilización de la especificación por todas las comunidades de práctica involucradas y no únicamente por el sector más tecnológico
- Son necesarios editores o herramientas de autoría con un diseño gráfico y un diseño de información más centrados en el usuario final y menos en los logros técnicos. Es cierto que este último año hemos pasado de no contar con ninguna aplicación basada en IMS-LD a tener más de una docena centradas o en torno a IMS-LD, lo que supone

un gran avance; pero el punto en común de todas ellas es la realización técnica de lo que la especificación define, no la utilización sencilla por usuarios reales. Es decir, con las aplicaciones informáticas actuales se pueden construir Unidades de Aprendizaje, mayormente sencillas o en nivel A, pero el grado de conocimientos técnicos necesarios es alto y la facilidad de uso de los interfaces es escasa. Es necesario un interfaz con un mayor grado de usabilidad, con una metáfora de aplicación y un sistema de diseño gráfico drag&drop, además de una ayuda contextual y bien documentada, en vez de un sistema de solapas y etiquetas para rellenar sin mayor soporte ni información y sin una conexión educativamente metodológica entre ellas, que es lo que existe en la actualidad

- Juntando la interoperabilidad, con la creación de patrones de aprendizaje y el desarrollo de editores visuales, los profesores y pedagogos podrían modelar Unidades de Aprendizaje centradas en aspectos más específicos, como el aprendizaje personalizado o adaptativo, que constituye uno de los primeros objetivos de IMS-LD, el aprendizaje colaborativo, la evaluación, la integración de juegos o el seguimiento de los usuarios en tiempo real [61], por citar algunos
- En referencia a los editores técnicos IMS-LD se necesita una evolución que permita trabajar de manera conjunta con el manifiesto, con los recursos y con los ficheros externos XHTML relacionados, mostrando las dependencias entre ellos y un sistema sencillo de edición y configuración de las propiedades
- La existencia de una capa de comunicación y el desarrollo de la interoperabilidad facilitaría la relación de IMS-LD con módulos y aplicaciones desarrollados con otras especificaciones (IMS Content Packaging, Scorm) otros lenguajes (PHP, Java) y con otros sistemas (Lams, Moodle). Uno de estos grupos lo constituyen los campus virtuales (Learning Management System –LMS- o Virtual Learning Environment –VLE-) desarrollado bajo cánones de IMS-LD o compatible con Unidades de Aprendizaje creadas en IMS-LD, con posibilidad de intercambio (importación y exportación) y ejecución y una capa de servicios (foros, chats, administración de usuarios, seguimiento de expediente, trabajo colaborativo, servicio de noticias o correo) que permita interactuar con ellas

2.11 . Justificación de la interoperabilidad y del aprendizaje adaptativo como dos de los principales retos actuales

Ya hemos argumentado la importancia de la adaptación y de la integración de Unidades de Aprendizaje como base de un aprendizaje personalizado y de un enfoque de colaboración abierta entre entidades de aprendizaje. En los dos próximos capítulos, extenderemos esta argumentación y la apoyaremos con investigaciones y datos concluyentes. Pero también queremos apoyar la importancia estructural de ambos temas en el futuro inmediato de la investigación y del desarrollo.

Por una parte, la especificación IMS-LD señala en uno de los tres documentos base que la componen (Information Model, Conceptual Model, Sección 2.1) [51] a la flexibilidad pedagógica, la personalización y la interoperabilidad como tres de sus objetivos base (R2, R3 y R6, respectivamente). Del mismo modo, otros objetivos se encuentran muy relacionados, como el R1, que promueve "el apoyo a una amplia variedad de enfoques de aprendizaje", o el "*R8. Reusability*", que se centra en la reutilización de elementos de aprendizaje en otros contextos. Dentro de las iniciativas de IMS-LD se encuentra la integración con otras especificaciones relacionadas, como IMS-QTI, IMS Enterprise, IMS Common Cartridge y IMS Vocabulary, pero también el diálogo con otros sistemas, como Moodle, LAMS, MOT+ y LAOS, que ya se está produciendo, y en donde ambos temas son la base del entendimiento.

Por otra parte, el Séptimo Programa Marco de la Comisión Europea [71], que regirá la investigación Comunitaria para los próximos seis años, contiene numerosos objetivos y definiciones en diversas convocatorias y retos que hacen referencia directamente a la adaptación y a la interoperabilidad. En concreto, sobre **adaptación y aprendizaje personalizado**, los objetivos 4.1 y 4.2 enfatizan la importancia de los sistemas de aprendizaje adaptativos e intuitivos, así como de contenido interactivo y expresivo que se exprese de manera no lineal. Extracto textual:

– ICT-2007.4.1 Digital libraries and technology-enhanced learning

> ***Adaptive and intuitive learning systems****, able to learn and configure themselves according to their understanding and experience of learners' behaviour. Cross-disciplinary research on the synergies between learning and cognition in humans and machines should lead to systems able to identify learner's requirements, intelligently monitoring progress, capable of exploiting learners' abilities in order to let them learn better, and able to give purposeful and meaningful advice to both learners and teachers either for self-learning or for learning in a collaborative environment.*

– ICT-2007.4.2 Intelligent content and semantics

> *Advanced authoring environments for the creation of novel forms of* ***interactive and expressive content enabling multimodal experimentation and non-linear story-telling****. These environments will ease content sharing and remixing, also by non-expert users, by automatically tagging content with semantic metadata and by using* ***open standards*** *to store it in networked repositories*

supporting symbolic and similarity-based indexing and search capabilities, for all content types.

Architectures and technologies for ***personalised distribution, presentation and consumption of self-aware, adaptive content****. Detecting and exploiting emergent ambient intelligence they will use features embedded in content objects and rendering equipment to enable* ***dynamic device adaptation, immersive multimodal experiences and contextual support of user goals and linguistic preferences****. Privacy preserving learning algorithms will analyse user interactions with devices and other users so as to update and effectively serve those goals and preferences.*

Y de manera más indirecta, aunque relacionada, encontramos argumentos en el Objetivo 1, y más concretamente en el 1.2 que apunta a la definición personalizada y dinámica de servicios. Del *Challenge 1: Pervasive and Trusted Network and Service Infrastructures*, citamos:

- Objective ICT-2007.1.5: **Networked Media new forms of experiences for individual users or user communities**.
- Objective ICT-2007-1.3: ICT in support of the networked enterprise

  Tools and technologies that enable ***intra-entreprise collaboration*** *and the definition and execution of tasks and workflows for operation across multiple domains.*

- Objective ICT-2007.1.2: Service and Software Architectures, Infrastructures and Engineering

  Service architectures, platforms, technologies, methods and tools that enable contextawareness and discovery, advertising, ***personalisation and dynamic composition of services****.*

A su vez, sobre **interoperabilidad**, encontramos diversas referencias explícitas en varias convocatorias y objetivos prioritarios no únicamente centrados en mejorar y ampliar las facilidades tecnológicas, sino en la aplicación de la misma a salud y medioambiente, por ejemplo. Objetivos como el 1.1, 1.3, 1.5, 3.3, 5.1, 6.3 y el 9.2 defienden la interoperabilidad y estandarización de sistemas y herramientas:

- Objective ICT-2007.1.1: The Network of the Future

  Ubiquitous network infrastructures and architectures supporting: i) convergence and ***interoperability*** *of heterogeneous mobile and broadband network technologies*

- Objective ICT-2007-1.3: ICT in support of the networked enterprise

Generic integrated solutions for ***inter-enterprise interoperability and collaboration*** *in the context of the networked enterprise.*

- Objective ICT-2007.1.5: Networked Media

  Roadmapping and conference support, for co-ordination with related national or regional programmes or initiatives, ***for international standardisation and interoperability initiatives****.*

- Objective ICT-2007.3.3: Embedded Systems Design

  Suites of ***interoperable design tools*** *for rapid design and prototyping: (1) increased interoperability of tools from SME vendors*

- Challenge 5: Towards sustainable and personalised healthcare

  The integration in healthcare processes and ***the interoperability of eHealth systems*** *should be part of the design and validation of the proposed solution.*

- Objective ICT-2007.6.3: ICT for Environmental Management and Energy Efficiency

  Specific International Cooperation Action (SICA) in ICT for environmental disaster reduction and management, the assessment of natural hazards and communities vulnerability together with the ***development and interoperability of rapidly deployable ICT-based solutions*** *for public warnings and emergency management.*

- Objective ICT-2007.9.1 (ICT-2007.9.2): International cooperation

  Open Source Software with particular focus on Asia, ACP and Latin America. To promote ***interoperability and the emergence of global open standards*** *and practices.*

Adicionalmente, podemos encontrar en el mismo documento casi cuarenta referencias sobre estandarización, contemplando por lo tanto, las especificaciones.

Similar atención reciben estos temas en otras fuentes nacionales de financiación, como la holandesa, la australiana, la canadiense o la estadounidense, con el consorcio permanente IMS y con el Departamento de Defensa de Estados Unidos buscando una solución mantenida a SCORM.

Por otra parte, existe un gran número de proyectos, grupos de investigación y de iniciativas, extendidos ampliamente por todo el mundo, que se encuentran centrados en el desarrollo de soluciones y extensiones sobre estándares, haciendo suyos los objetivos descritos (.LRN, TENCompetence, ProLearn, EU4ALL, LAMS, Moodle, Licef, Bolton, etcetera) [47].

De todo esto se deduce la gran relevancia actual tanto de la adaptación como de la interoperabilidad y la implicación con estándares. Pero también el empuje y la presencia que en el próximo lustro tendrán en Europa y en otros focos internacionales de desarrollo.

En esta sección hemos analizado la estructura y los elementos generales de IMS-LD, obteniendo unas conclusiones parciales sobre aspectos a modificar y a añadir. Destacamos la escasez de estructuras de programación y la laboriosidad de modelado asociada al *information model* y a la definición propia de la sintaxis. Colateralmente, y aunque no se aborda en la solución desarrollada en esta tesis, también subrayamos la ausencia de herramientas visuales de alto nivel que permitan una penetración mayor de IMS-LD en los públicos objetivo potencialmente usuarios de la misma.

En la sección siguiente profundizaremos en la implementación específica de la integración de UoLs en diversos contextos, tales como la integración con otras especificaciones (SCORM), con LMSs (Moodle) y con objetos de aprendizaje (interna y externamente).

3. Interoperability and integration of Units of Learning

While previous section concentrates on general aspects of IMS LD (i.e., structure, elements, levels), this section is focused on one of the two main lines of research in this thesis: interoperability and integration. We depict this integration along four different scenarios: integration with other eLearning specifications (e.g., SCORM), integration with LMSs (e.g., Moodle), integration with Learning Objects internally modelled, and integration with Learning Objects externally modelled (e.g., eGames, in both cases).

3.1 . A general research framework: common misconceptions on IMS-LD, interoperability and integration

There a number of common misconceptions on IMS-LD and interoperability. They come from a lack of knowledge from the users but also from a drawback on the dissemination activities. Although IMS-LD went public in 2003, still there is a blur approach to what it is, what it can do and how. After more than twenty workshops in the last three years in several countries, the author has detected several misunderstandings. They come from private and informal conversations between the participants and also with the teacher, so no library reference is provided. However, they are meaningful and should be considered as a first-hand experience from the field, since they become enlighting about what a wide variety of users think. The ten most often misconceptions are:

1. IMS-LD can be compared to any LMS (Learning Management System) because IMS-LD aims to do the same things than any LMS. It's usual to think that IMS-LD is also a Course/Learning Management System
2. IMS-LD is less powerful than any LMS because with any LMS you have several services included and they are very easy to incorporate. IMS-LD has no service and there is no way to incorporate it. Where are the services in IMS-LD?
3. IMS-LD is less efficient than any LMS because the Units of Learning cannot be changed on-the-fly, at run time. The division between design-time and run-time are seen as a defect in IMS-LD
4. Any LMS and IMS-LD can develop the same kinds of pedagogical approaches. Any LMS is powerful enough to replicate the pedagogic approach of IMS-LD and IMS-LD is basically so constructivist as any LMS (for instance, Moodle)
5. IMS-LD is not as complete as any LMS because, in any LMS, you can create Courses dragging and dropping and with IMS-LD you need to use not-friendly

editors. So, the misconception is that IMS-LD itself is used as an Editor, this is, as a tool

6. Any LMS is better for teachers because you can create courses in an easy way and keep focused on didactical issues as long as you have to know a lot of technical issues in IMS-LD and loose the pedagogy (for instance, Lams). So, you need to know all the Spec to start creating didactical material instead of starting straight ahead

7. IMS-LD is a well-established standard and not a specification. In fact, there is no difference between a standard and a specification

8. IMS-LD fights against all its possible critics and locks the publication and dissemination of them, avoiding the discussion. Besides, wealthy companies are pressing quite a lot for IMS-LD to be adopted in order to make money with their companies

9. CopperCore is a LMS and it is comparable to any LMS. So, CopperCore is a really bad LMS because it cannot provide the same facilities and modules than any LMS

10. IMS-LD is worse than any LMS because IMS-LD needs an engine and a specific browser and any LMS can be run in a simple generic browser and it doesn't need any engine. In general, people don't know what CopperCore is and what it is for

In addition to the lack of knowledge and dissemination, these ten misconceptions stress what the users do to evaluate to compare it with their reference models, any LMS in this case. As they normally use and know about LMS (Claroline, Moodle, .LRN, Lams…) they take them as a base to achieve IMS-LD.

However, LMS´s are not the only one reference that users have. In the coming subsections we deal with the usual resources that teachers and learners use in the learning process, in regards with interoperability and integration of Units of Learning, taken as information packages with resources and supporting learning processes. In addition, in order to focus our research we have selected the most prominent examplar out of each category:

a) other eLearning specifications to model the courses and learning packages. We use Scorm, as the historical reference specification on eLearning

b) a Learning Management System to deliver the learning packages and provide a layer of services, such as forums, chats, news or whiteboards. We use Moodle, as the reference and most raising LMS in the open source world

c) learning objects to provide content and resources to the learning packages, internally and externally. We use eGames and simulations, as one of the learning objects (LOs) with a higher level of interaction within the learning flow

All of them aim at interoperability-integration as one of the main objectives in order to be implemented and to evolve. All of them will be supported and justified subsequently and they will always be related and matched to the main topic of this dissertation, IMS Learning Design.

3.2 . Integration with another eLearning specification: IMS Learning and SCORM

Standardisation plays an increasingly important role in eLearning, requiring designers to make choices as to the standardisation route to be followed during the development of eLearning courses [72]. IMS Learning Design is an eLearning specification which allows eLearning designers to describe Units of Learning – delimited pieces of education or training, such as courses, modules or lessons. SCORM 2004 is the latest version of Advanced Distributed Learning's reference model for eLearning, which describes a content model and run-time environment for Shareable Content Objects. IMS Learning Design and SCORM 2004 are often positioned as mutually exclusive alternatives. This section outlines the case for using the two together and examines approaches to achieving integration between Units of Learning and Shareable Content Objects. The section concludes by proposing that a future version of the SCORM should integrate IMS Learning Design to consolidate the eLearning standardisation progress made to-date.

3.2.1. Introduction

There is a significant amount of interest in standardisation in eLearning, signalling a growing maturity in the field [73-75]. However, the standardisation area, which includes de jure standards and de facto specifications [76] is difficult to oversee, and several guides have been produced to help practitioners understand the terrain [77-79]. Nevertheless, users of eLearning specifications and standards continue to point to their complexity [39; 80].

Two important pieces of the eLearning standardisation puzzle are IMS Learning Design [81] and the Sharable Content Object Reference Model [11]. While the two have different natures – IMS-LD is a single specification whereas SCORM 2004 is a reference model containing a number of specifications – their application areas and terminology overlap to a sufficient degree that confusion exists as to their relationship, and there is ongoing speculation in the eLearning community on when to use which one, and with which intended benefit [82; 83].

IMS-LD is a specification used to model Units of Learning (UoL) – "any delimited piece of education or training, such as a course, a module, a lesson, etc". A UoL goes beyond a collection of learning resources, representing the whole learning process, including learning activities (problem solving activities, search activities, discussion activities, peer assessment activities, etcetera), assessments, services and support facilities provided by teachers, trainers and other staff members [84; 85]. UoLs are content packages [86] containing materials fitting a prescribed structure which, as a result, can be 'played' by software able to interpret the materials, such as the CopperCore engine [87]. Although a single specification, IMS-LD is designed to orchestrate learning arrangements, and so is linked to several other eLearning specifications (see section 5.3 of the IMS-LD Best Practice and Implementation Guide, part of [81] for details).

SCORM 2004 is the latest version of the Sharable Content Object Reference Model (SCORM), consisting of a Web-based learning Content Aggregation Model (CAM), Run-Time Environment (RTE) and Sequencing and Navigation behaviour for learning objects [11]. If educational material is created according to the SCORM 2004 model, a SCORM 2004 compliant RTE will be able to 'play' the material and the expected run-time behaviour will result. The SCORM 2004 requirements on the content cover not only its structure and packaging but also requirements on implementing run-time behaviour so that communication between a running Shareable Content Object (SCO) and an associated Learning Management System (LMS) is facilitated. Mackenzie [88] provides a good introduction to SCORM 2004 and Ostyn [89] takes a more detailed look at the workings of an earlier version of the reference model.

These two brief descriptions illustrate similarities between IMS-LD and SCORM 2004 – both can be used to guide the development of educational materials, both use a combination of specifications to achieve their goals and both lead to content packages which can be read into players and used to support learning.

However, these similarities mask fundamentally different views on learning. Several authors have pointed out that SCORM is currently centred on a single learner model [90-93] while IMS-LD allows learning flows involving groups of learners to be represented. Furthermore, IMS-LD is able to model learning experiences involving multiple roles (eg tutor, learner, coach) and, drawing on the constructivist movement, places learning activities rather than learning objects at the heart of its model [54]. With respect to this latter point, a significant amount of confusion has been generated by IMS Learning Design and SCORM both using the term 'activity', but with different meanings:

"Activities are one of the core structural elements of the 'learning workflow' model for learning design. They describe the activities a role has to undertake within a specified environment composed of learning objects and services." [81]

"an Activity may be a learning resource, but it can also be a collection of sub-activities (each of which may be either a collection of further sub-activities or a learning resource)." [88]

SCORM 2004 takes a resource-centred view of activity, whereas IMS-LD uses the term to specify what someone in a particular role must do to achieve learning objectives.

Despite the fundamental differences and terminological difficulties, the two can be used together to give a useful eLearning combo. The following section first provides the motivation for combining IMS Learning Design and SCORM 2004. This is followed by a section which examines increasing levels of integration in detail and the section concludes with some remarks on a future, perfect marriage between the two.

3.2.2. Why and how to use IMS Learning Design and SCORM 2004 together

UoLs typically incorporate content to help learners and/or staff carry out their activities, and this content can be in a variety of formats, including XML, (X)HTML, RTF, PDF etc. In contrast to its predecessor EML [29], IMS-LD does not prescribe a model to which content must adhere but focuses instead on specifying the learning process in terms of which roles perform which activities, when, and supported by which facilities. As a result, IMS-LD provides a natural slot into which SCOs can be incorporated, and existing SCOs could be re-used in this way. SCORM 2004 is the third main release of the SCORM, which has been in use for 4-5 years. In this time the reference model has been used by content developers to create SCOs and the existing corpus of SCOs could be leveraged. We note, however, that numbers of existing SCOs are hard to come by (and a Google search on scormtype="sco" returns surprisingly few results), though perhaps given the military roots of the SCORM, many SCOs are not publicly available.

Following the approach of incorporating SCOs within UoLs gives a solution to the "single learner, single role" shortcomings of SCORM 2004 noted above so that, at some point(s) in a learning process, learners carrying out learning activities are able to turn to SCOs to support their learning. In fact, SCOs can be made available to other roles to help improve their competencies, such as supporting tutors with hints and tips on giving online feedback to students.

Integrating SCOs into UoLs is a question of placing the SCORM 2004 content in the context of one or more learning objects in one or more environments in an IMS-LD UoL. The next section explores different ways of approaching this issue.

3.2.3. Approaches to Integration of IMS Learning Design and SCORM 2004

Ostyn [89] draws the distinction between minimal SCOs and data-enabled SCOs. This distinction can be used in combination with IMSCP's external resource referencing capability to allow different levels of integration between IMS-LD and SCORM 2004.

3.2.3.a. Minimal integration

Minimal integration involves simply referencing, from within a Unit of Learning, a SCORM-based LMS running a SCO, and is illustrated in Figure 19.

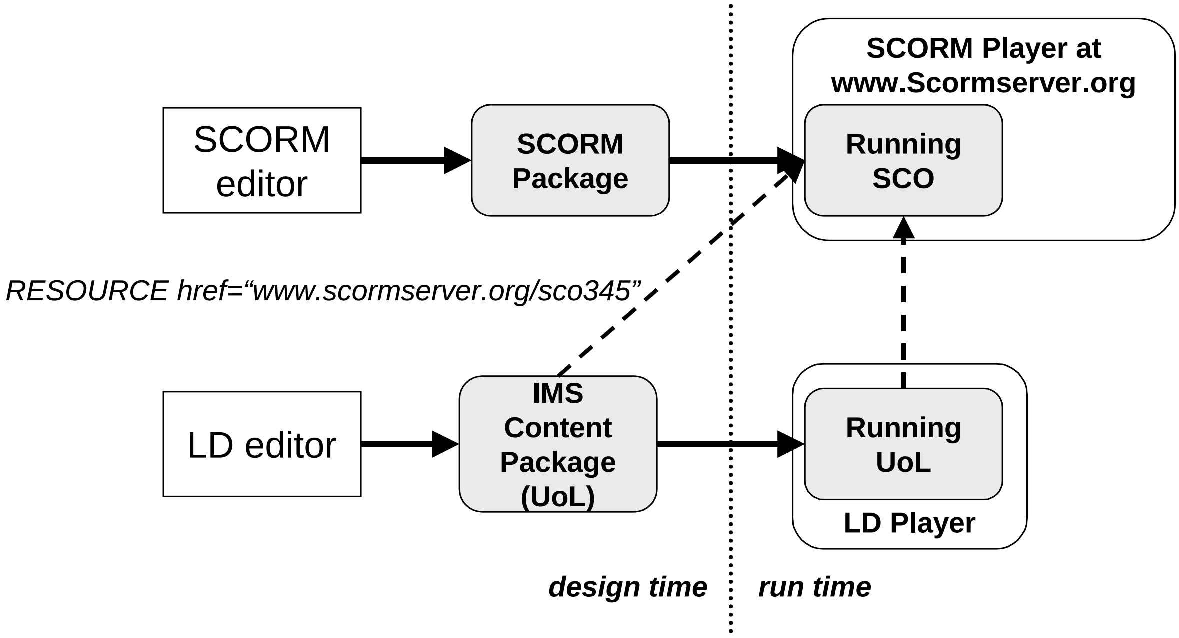


Figure 19. Minimally integrating a SCO in a UoL

During design time, a SCORM package containing one or more SCOs is authored using a SCORM editor and delivered into an environment which includes a SCORM 2004 aware run-time configuration. We refer to this as a SCORM Player, which may either be a full LMS, or, as is the case with the RELOAD SCORM Player [94], a stripped down single-user LMS. The SCORM Player is able to launch a SCO and handle the required SCO-LMS communication.

The SCORM Player is addressable through a URL (labelled as www.scormserver.org in the diagram), which can be combined with a SCO identifier to create a new URL through which the SCO can be launched. This URL can be used from within a UoL to reference the SCO. Figure 20 shows an example of this minimal level of integration using the player which accompanies the CopperCore IMS-LD engine [87]. The top-left-hand panel displays the title of a learning activity ("Read the resources contained in the environment"). The bottom-left-hand panel shows the environment associated with this learning activity (entitled “Link to running SCO”), which contains a single learning object “Follow me to Moodle”. Clicking on the learning object opens a new window and launches an example SCO running in the SCORM player available in Moodle [95].

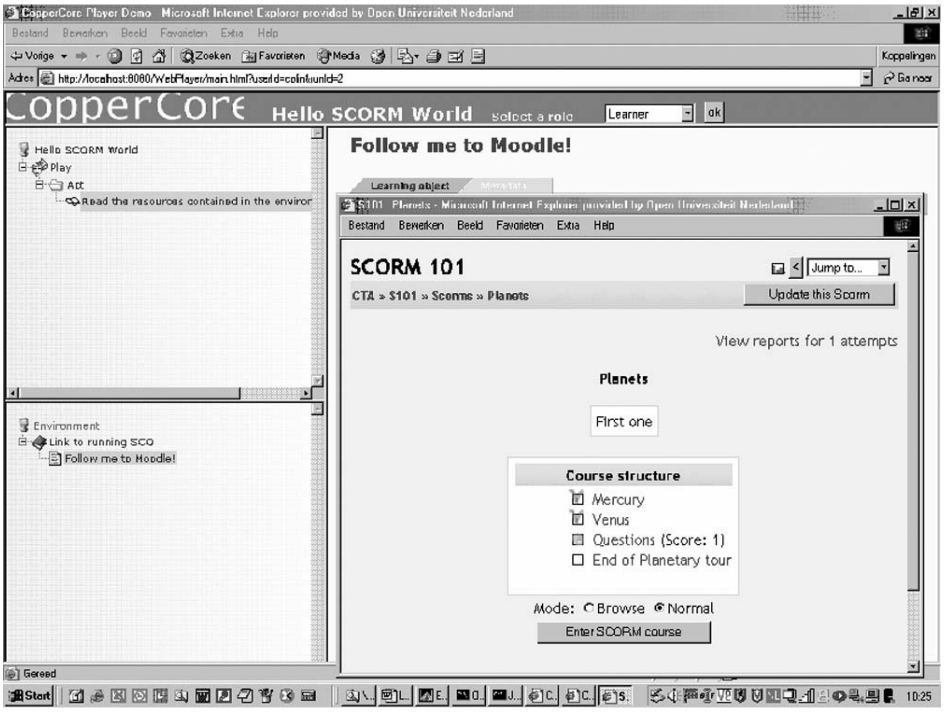


Figure 20. The CopperCore IMS-LD player referencing a SCO running in the Moodle SCORM player

Minimal integration has simplicity as its main advantage – it can be accomplished straightforwardly with existing tools and content. Although the UoL shown is a toy example, we can imagine learning situations in which individuals in a cohort of learners are invited to share and discuss their preconceptions on planetary motion before being provided with content individually on the planets for self-study. Once digested, a second collaborative phase guided by a teacher could be used to encourage the learners to reflect on changes in their understanding. All this is possible using IMS-LD concepts and software in conjunction with SCORM 2004 content and software.

Minimal integration does however suffer from some drawbacks. Learning designers may not know the final URL location of a SCORM course as they design a UoL. Moreover, URLs are notoriously short lived and the approach can lead to learners following dead links. The handle system presents a solution to the changing URL problem, and is under active consideration in the SCORM community [96].

Perhaps the bigger drawback is the lack of communication between the running UoL and the running SCO. This removes the possibility to have information on the learners interaction with the SCO be used to inform further progress of the UoL (eg if the SCO involves a test, the results might be used to couple the learner to peers with similar scores).

*3.2.3.b. **Packaged integration***

In order to avoid the need for learning designers to know URLs, the packaged integration approach exploits the fact that UoLs and SCOs are both packaged using IMS Content Packaging and that nesting of packages is allowed. Figure 21 sketches the approach.

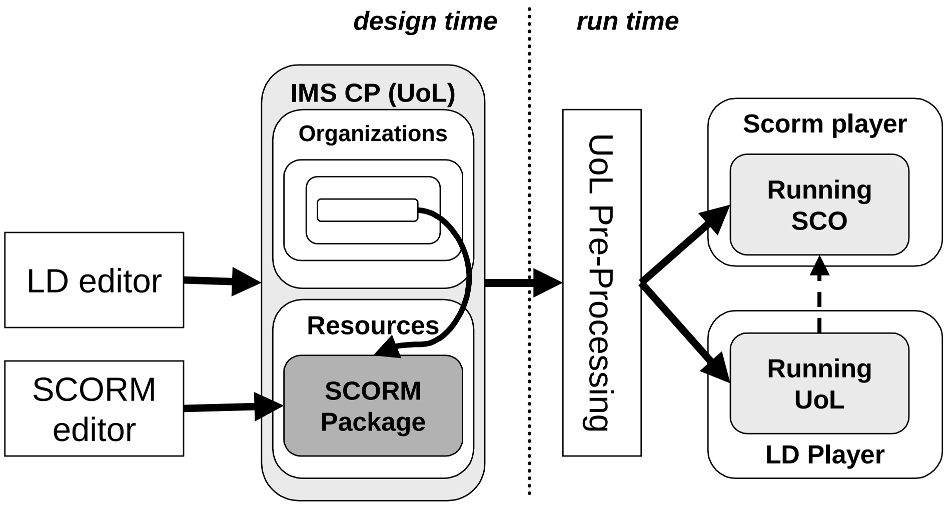


Figure 21. Embedding a SCORM package in a UoL

Again, editors are used to create a UoL and a SCORM package. However, the two are packaged together and instead of the UoL referencing content external to the package, the embedded SCORM package is used as the resource associated with the Learning Object. Figure 22 shows a IMS-LD Learning Object being edited in the RELOAD learning design editor [94] to point to a local SCORM package, for subsequent embedded packaging.

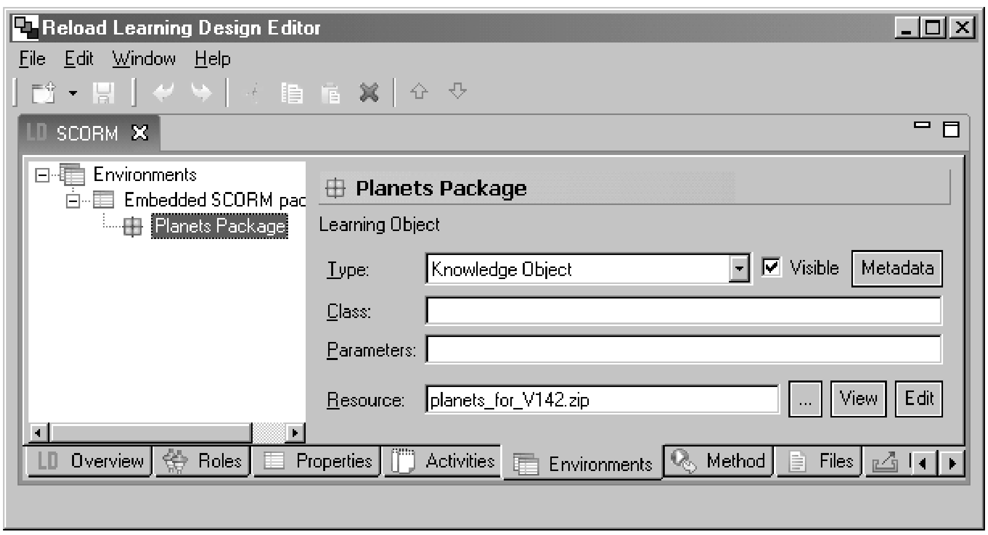


Figure 22. Using the RELOAD editor to embed a SCORM package in a UoL

Following this approach, the Package must be “disaggregated” into its constituent packages (one containing the SCO, the other containing the UoL) by some UoL Pre-processing mechanism. To avoid the need for this mechanism to examine the contents of the package in order to know how to process it, the IMSCP resource type should be set appropriately. Note that the set of possible values for the type attribute of an IMSCP resource does not include “scorm” and so an extension to include this value must be used here.

Once recognised and disaggregated, the SCORM package is then delivered to the SCORM Player, and the UoL is delivered to the learning design player, with a link from the UoL to the SCORM Player and SCO being generated on the fly. Figure 23 shows the RELOAD LD Player [97] with, in the right-hand pane, the RELOAD SCORM Player [94] running a SCO on the planets. The result shown in Figure 23 required manual intervention to simulate the UoL pre-processing step – this degree of integration, though relatively straightforward, is not yet available in tools.

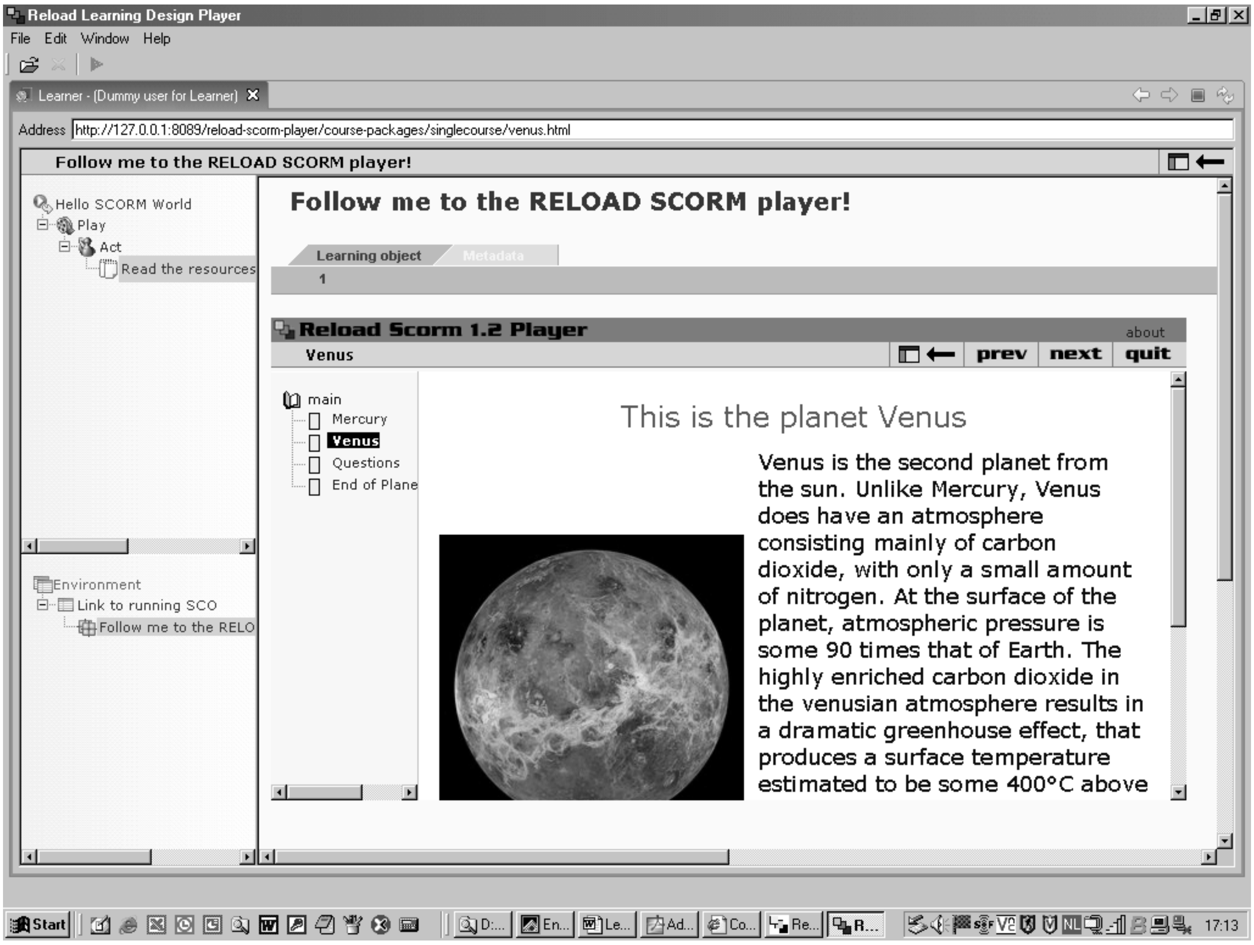


Figure 23. The RELOAD LD and SCORM Players interoperating with IMS-LD and SCORM content.

The packaged integration approach solves the URL problem, although from a re-use perspective, it has the negative consequence of leading to a proliferation of copies of SCORM packages (see Ip and Canale [98] for an approach to solving this issue). Moreover, the approach still has the drawback of excluding communication between the running UoL and the running SCO, limiting the benefits to be gained from the combination of IMS Learning Design and SCORM 2004.

3.2.3.c. ***Runtime integration***

If information on the learner's status and progress in a SCO were to be available to a running UoL, it could be used to influence the learning flow following completion of the SCO. In this way, the learning activities presented to a learner could vary depending on learner-SCO interaction, such as including additional remedial activities, or skipping parts of the learning flow which appear on the basis of tests to be within the learner's competency level. Moreover, the SCO run-time information could be used during the execution of a SCO to trigger events in the UoL. This opens a number of possibilities for collaboration within and around SCOs – we can imagine a chat facility being opened (a Learning Service in IMS-LD) between a tutor and a learner if the time spent on a particular SCO exceeds a certain threshold.

Such new possibilities do, however, increase the complexity of the situation significantly, requiring both a design time and run time interweaving of UoLs and SCOs. We first introduce the data items defined in SCORM 2004 and examine how UoLs could be designed to exploit this data to influence learning flows. We then examine the run-time situation to identify the software necessary to achieve a tight integration of IMS-LD and SCORM.

The SCORM 2004 Run-Time Environment Data Model specifies a set of data model elements which can be used to monitor SCO information. Table 2 lists shows which information is available to be exploited.

Table 2. SCORM Run-Time Environment Data Model Elements Summary

Data Model Element	Function
Comments from Learner	Contains text from the learner.
Comments from LMS	Contains comments and annotations intended to be made available to the learner.
Completion Status	Indicates if the learner has completed the SCO.

Completion Threshold	Identifies a value against which the measure of progress the learner has made toward completing the SCO can be compared to determine if the SCO should be considered completed.
Credit	Indicates if the learner will be credited for performance in the SCO.
Entry	Contains information to indicate if the learner has previously accessed the SCO.
Exit	Indicates how or why the learner left the SCO.
Interactions	Defines information pertaining to an interaction for the purpose of measurement or assessment.
Launch Data	Provides data specific to a SCO used for initialization.
Learner Id	Identifies the learner for whom the SCO instance was launched.
Learner Name	Represents the name of the learner.
Learner Preference	Specifies preferences associated with the learner's use of the SCO.
Location	Represents a location in the SCO.
Maximum Time Allowed	Indicates the amount of accumulated time the learner is allowed to use a SCO in the learner attempt.
Mode	Identifies the modes in which the SCO may be presented to the learner, such as "browse," "review" and "normal."
Objectives	Specifies learning or performance objectives associated with a SCO.
Progress Measure	Identifies a measure of the progress the learner has made toward completing the SCO.
Scaled Passing Score	Identifies the scaled passing score for a SCO.
Score	Identifies the learner's score for the SCO.
Session Time	Identifies the amount of time that the learner has spent in the current learner session for the SCO.

Success Status	Indicates if the learner has mastered the SCO.
Suspend Data	Provides information that may be created by a SCO as a result of a learner accessing or interacting with the SCO.
Time Limit Action	Indicates what the SCO should do when the maximum time allowed is exceeded.
Total Time	Identifies the sum of all the learner's session times accumulated in the current learner attempt prior to the current learner session.
Version	Identifies the SCORM implementation version.

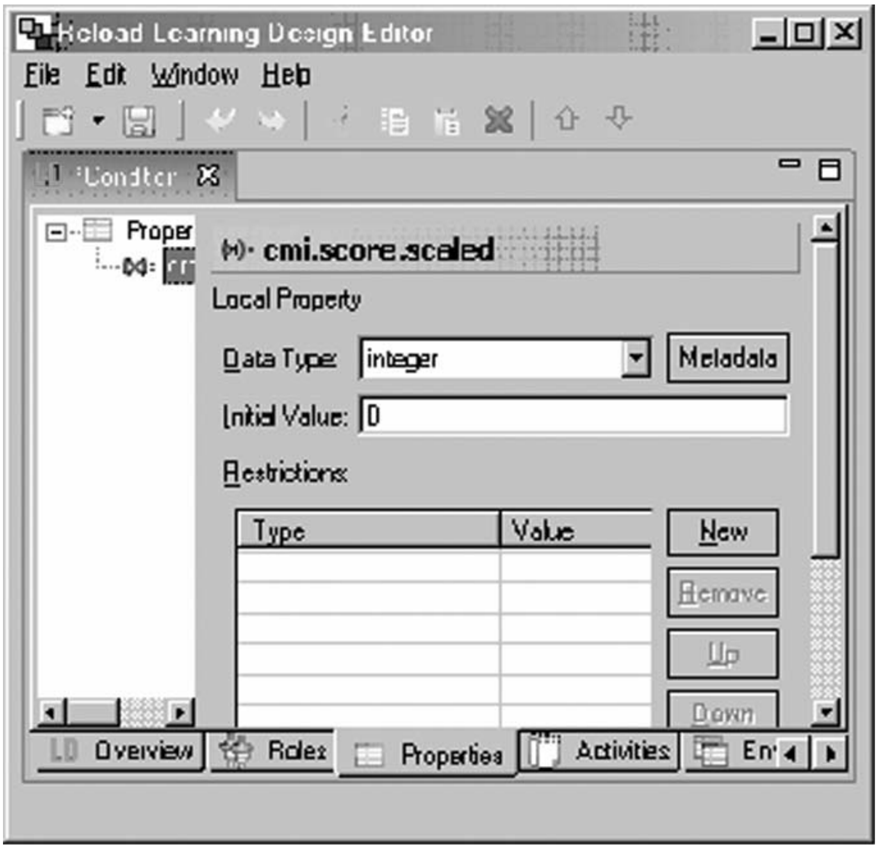
SCORM 2004 also specifies a so-called Application Programming Interface through which standardised communication between a SCO and an LMS can occur. Essentially, the SCORM states which instructions are allowed in communication. In combination with the data elements shown above, information on, for example, the learner’s score in a SCO can be communicated to the LMS through a SET instruction using a prescribed syntax:

```
SetValue(“cmi.score.scaled”,”0.75”).
```

Leaving aside the mechanism by which the score is calculated, the above statement can be interpreted as meaning that the learner’s score on the SCO is 75%.

How might this information be used in a UoL? A large part of IMS-LD's flexibility in orchestrating learning flows comes from its use of properties and conditions. Properties are used to record various types of information, which can be used in conditions to influence aspects of the learning process, including the ordering and visibility of learning activities and learning objects. In a typical example, learning designers might create a property to store the results of a test. This property could then be used in a condition to reveal a previously hidden activity. The following figure (Figure 24) shows the creation of a property and a condition using the RELOAD Learning Design Editor.

The property is declared, given a type (integer) and an initial value (0) …..



…. then used in a condition which states that if its value is greater than 75, a previously hidden learning activity should be shown to the learner.

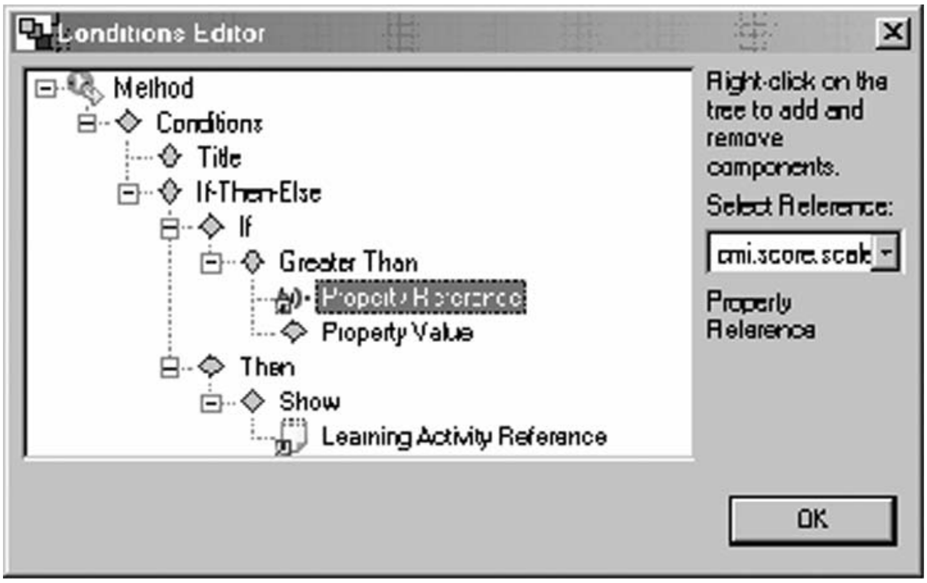


Figure 24. Creating IMS-LD Properties and Conditions with the RELOAD LD editor

The route to tighter integration involves learning designers defining properties and conditions which refer to the SCORM 2004 data model elements supported by data-enabled SCOs (we note here that since a UoL could involve more than one SCO, a property naming convention needs to be agreed to uniquely identify SCOs). This does, of course, place rather a heavy burden on designers, who need to be acquainted with both

IMS Learning Design and SCORM 2004. Integrated tool support in which the data elements maintained by individual SCOs are able to be dragged and dropped into IMS-LD conditions, creating IMS-LD property definitions on the fly, would seem to be essential.

The above shows how learning designers are able to create UoLs but also raises requirements on the run-time context in which integrated UoL-SCO combinations are delivered. It is no longer possible to simply hand off control from a UoL to a SCORM Player when the learner clicks on a link; instead, the environment in which a UoL runs must be extended to include the capability to launch and communicate with a SCO and where necessary to map and synchronise SCORM 2004 data elements and IMS-LD properties. Figure 25 shows the integrated situation.

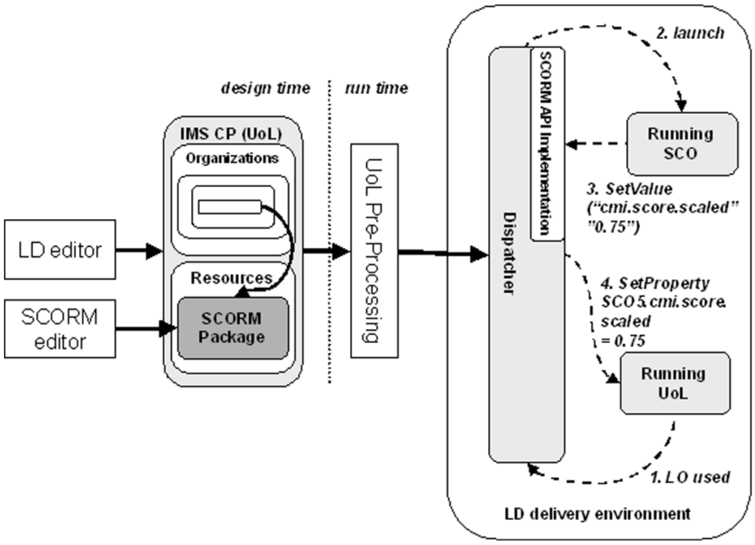


Figure 25. A deeper level of integration

Figure 25 introduces a new piece of functionality known as the Dispatcher. This is a central coordination and synchronisation mechanism responsible for ensuring that the right content is made available to learners and staff at the right time and that data is shared and mapped between underlying components appropriately. If, in addition to containing a SCORM package, the UoL also contains a learning flow condition which uses data from a running SCO, we can trace the sequence of events in the IMS-LD delivery environment to illustrate the functionality involved. When the Learning Object is used (step 1) a request is sent to the Dispatcher to Launch the relevant SCO (step 2). The Dispatcher acts here as an LMS, following the requirements for LMS-SCO initialisation and communication described in SCORM 2004. The learner interacts with the SCO, which at some stage in the interaction issues a call to set the value of cmi.score.scaled (step 3). The Dispatcher implements this functionality and also sets the corresponding IMS-LD property (step 4). This property is used in an IMS-LD condition, which is triggered by this change in value, causing a new activity to be shown in the running UoL.

3.3 . **Review of IMS-LD based on the integration with SCORM**

This section has examined why and how IMS Learning Design and SCORM 2004 should be seen as complementary rather than alternative routes to creating eLearning courses. It follows the natural division of using IMS Learning Design to describe the orchestration of learning processes into which SCORM 2004 content is slotted at appropriate points. Learning designers benefit from the combination by being able to tap into the existing body of content developed according to the SCORM model (indeed, the use of SCORM is mandated in some contexts) while at the same time being liberated from the single learner, single role model through IMS-LD's broader pedagogical scope.

In addition to implementing the approach described in this section, our further work will also take into account the sophisticated sequencing and navigation capabilities available within SCORM 2004 [99], seeking to ensure separation of concerns so that specialized SCORM 2004 players allow SCOs to be viewed as black boxes by the Dispatcher.

More fundamentally, much of the complexity of combining UoLs and SCOs would be removed if IMS-LD were to be included in a future version of the SCORM. Such a step would align the data models, run-time behaviour, terminology and packaging, marrying the best of both and presenting eLearning practitioners with a consolidated eLearning reference model for the coming generation of eLearning content and systems.

IMS-LD was designed with such an integration in mind, and provides a placeholder in the 'environment' associated with an activity. Figure 26 shows this arrangement.

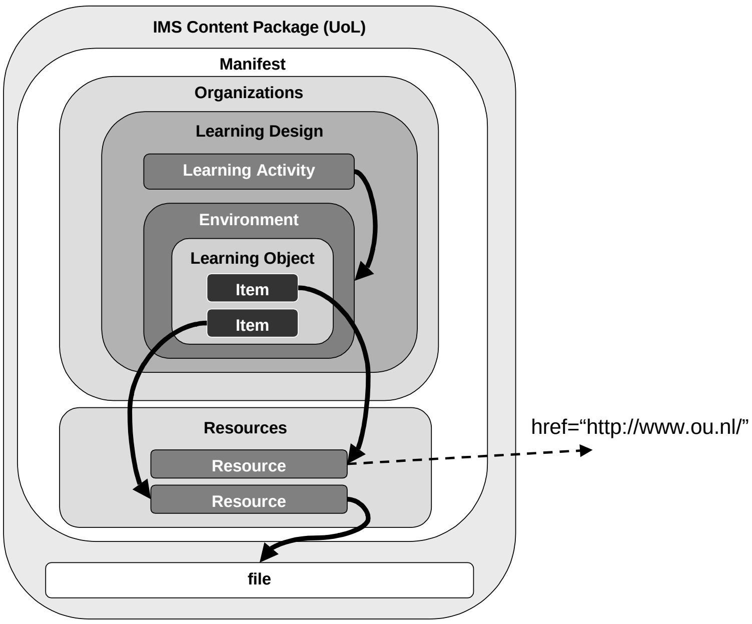


Figure 26. Key concepts from IMS Learning Design

A UoL is an IMS Content Package including a manifest, a learning design, resources and physical files. The learning design is included as an organization within the IMSCP

<organizations> element. IMS-LD defines a learning object as “any reproducible and addressable digital or non-digital resource”. The Learning Activity shown in Figure 26 is associated with an environment containing a single Learning Object, which contains two items. Each item, in turn, points to a resource in the resources section of the content package. The figure illustrates that IMSCP resources may reference files included in the package or content outside the package but available through a URL.

In the same line as IMS-LD and SCORM or IMS-LD and IMS Content Packaging, the relationship between two or more different specifications look at the same levels of integration. Furtheron, we address a different problem with non-standarized learning objects and similar solutions, from a minimal integration to a run-time/deeper integration using a communication dispatcher between both entities.

3.4 . Integration with a Learning Management System: IMS-LD and Moodle

Integrating the specifications and tools for IMS-LD into Moodle [10], an open-source Learning Management System (LMS), is not just a technological question, but also relates to practical, pedagogical, and philosophical issues. This section documents the discussions and experiments of a team of teachers active in the Moodle community who are concerned with the development of international standards in future versions of Moodle [100]. Participants analysed the implications of integrating the IMS-LD specification into Moodle and the operation of various IMS-LD tools (Coppercore, Reload) and related tools (LAMS, eLive) around the Moodle environment. These differences were then summarized into general implications for future versions of both Moodle and IMS-LD. This section concludes that continued, open dialogue between teachers and developers of both IMS-LD and Moodle is necessary to achieve transparent integration [100].

3.4.1. Introduction

The creation of any international standard is a complex endeavor, and most particularly in a practice-based craft such as teaching and learning. Educational stakeholders have an interest in a common method of exchange across borders, languages, codes, venues, methods, philosophies, and interfaces. Until now, however, this exchange has been limited to printed materials, a costly and physically limiting media. Early attempts at digital standards have focused on narrow areas such as quiz question packaging or sequenced content. Yet, teachers in particular are hungry to share full courses and learning scenarios, complete with content and processes that they have found useful. The IMS-LD specification (IMS-LD) is one attempt to bring that fuller picture to electronic exchange that can theoretically include all forms of highly complex and flexible learning for both online and face-to-face learning venues.

Traditionally, the design of pedagogy has been the realm of expert instructional designers, textbook authors, and software engineers. With the advent of easy-to-program web scripting languages and simplified digital authoring software, teachers are playing a greater role in the creation of learning materials and designs. Furthermore, the popularity of open source course management systems with pluggable modules and point-and-click configuration has allowed teachers to experience unprecedented freedom of design. Now these teachers want to share learning objects or learning units with each other, first in teams, then across departments, and now amongst any institution using any kind of system. That is the emergent demand which leads to their interest in international standards. As the role of teachers grows, we see other stakeholders such as engineers, academics, developers and IT professionals playing a comparatively less directive, but more supportive role in the co-creation of these standards. Educators active in the Moodle community are especially interested to join in this dialogue with the IMS-LD community. A final stakeholder, the learners, is not the main subject of this section, but we acknowledge their growing role and responsibility in the design of learning.

Moodle is an open source learning management system (LMS) that has maintained interest in the IMS-LD specifications over the past two years since community discussions began on this topic. At that time, it was noted that the IMS-LD specification was the most congruent standard for Moodle, since it allowed for learning scenarios to be constructed as sequences of learning activities rather than being restricted to sequences of learning contents or objects. Although Moodle can be used for many kinds of educational applications, it is based on socio-constructivist principles and most suited for an educational approach involving interaction amongst people rather than transmission of content. Furthermore, the PHP scripting and modularity of Moodle even allows teachers to supervise the creation of new activity tools in the LMS—the emergence of the teacher-developer.

This section represents a view not from Moodle programmer-developers, but from these teacher-developers who actively influence the application and pedagogical directions of Moodle. The overarching presumption we hold is that any IMS-LD process must be intuitive and empowering for teachers, and not intended solely as the professional realm of instructional designers. Our primary aim is to discuss the pedagogical and philosophical aspects of the process of moving to an international specification, IMS-LD, and secondarily to illustrate that process with our initial testing of IMS-LD tools. Our research questions addressed in this section are threefold:

- If IMS-LD becomes the standard for the design and exchange of teaching tools and materials in Moodle, how will it affect teachers accustomed to the current design approach in Moodle?
- What are the attributes of the current Moodle way of design that we wish to preserve?
- What are some strategies for integrating IMS-LD into Moodle?

We do not reach definitive conclusions, but summarize our collective impressions in this section. The sections of the section include: 1) describing our method for investigation, 2) outlining some difficulties in understanding the relationship between IMS-LD and Moodle, 3) exploring IMS-LD and IMS-LD-related tools in a Moodle environment, 4) drawing implications for future Moodle versions, and 5) making similar implications for future development of the IMS-LD specification and tools.

3.4.2. Method for Studying IMS-LD

In February of 2005, concurrent with the publication of the first book on IMS-LD [52], the Moodle community began an online study [101] of this book using a chapter-per-week format facilitated by members of the community with extensive experience using Moodle in formal and informal education. A separate course was set up on the moodle.org community site, and members were invited to join as either facilitators or participants. Ten members volunteered to serve as facilitators and another fourteen self-enrolled by contributing to the forum, “Why are you interested in Learning Design?”. The group represented a diverse background including secondary and tertiary level teachers from Australia, Vietnam, Spain, Netherlands, Venezuela, Japan, Germany, United States, Italy, New Zealand and Sweden. Occasionally, developers of Moodle and IMS-LD tools visited and offered comments. Eleven chapters were initially chosen based on relevance to pedagogical issues, rather than tool design. Facilitation of the discussions were based on the focus group protocol, a semi-structured methodology in qualitative inquiry [102]. Focus group discussion is useful when a research question into social phenomena cannot be clearly identified, requiring exploratory investigation to determine relevant issues [103]. The procedure followed in this case was for each facilitator to read the assigned chapter, then post 3-5 discussion questions for members to respond to. Over the course of the study, this process generated 21 forums, with over 200 topics and almost 2000 postings. In addition, each facilitator created a wiki (editable group document) to summarise their respective section. Finally, a moving wiki (repositioned week-by-week) was used to record notes by any course member on the implications of each section relative to Moodle and IMS-LD development. These notes and postings formed the basis for this summary document and the resulting commentary that has been subsequently incorporated. Collaborative writing by the seven person writing team was conducted both privately on a closed “Teachers Forum” and publicly in writing forums separated section-by-section.

3.4.3. Handling IMS-LD Concepts and Operation in the Moodle Environment

The initial immersion into IMS-LD gave us an experience of confusion over terms, concepts and tools. Our group constantly mixed discussions amongst conceptual points, codified specifications and multiple tools which are in various stages of development. Teachers will need to grasp these differences before a meaningful discussion can take place. This section begins a clarification of terminology and the functions, pedagogical descriptiveness, and styles of design such as bricolage.

3.4.4. **Functions and Terminology**

IMS-LD is a notation, a proposed standard for modeling learning scenarios, while Moodle is an LMS, a complete package for managing, designing and running courses. Thus the two do not compare directly, yet each uses a language to describe the process of designing a learning activity. The differences in the terminology are subtle and the absence of some concepts in each other's lexicon is a useful indicator that significant differences exist. For example, in the Moodle approach to design, the base structure is a "course", while in IMS-LD the principal term is a run of a "Unit of Learning" (UOL). A Moodle course includes user management, enrolment, learner monitoring, activity modules (tools), resources (attached files and links), all visibly arranged on a single main course page. In IMS-LD, the Unit of Learning is a packaged set of activities, roles, content, while Moodle has no direct equivalent (see discussion on pedagogical descriptiveness). While an IMS-LD UOL could be a whole course, in general, it is assumed that a number of UOLs will be assembled to make a full course. The assembling package is called an LMS, and IMS-LD does not directly attempt to model that total environment. Associated with IMS-LD are various firms and organisations which have developed tools called "editors" which create designs, and "players" which run them for students. In Moodle, those two roles are integrated in one environment. Other differences are illustrated in Table 3 and Table 4, a first attempt towards a dictionary that translates the differences used in the lexicons of IMS-LD and Moodle.

Table 3. Differences in Terminology for IMS IMS-LD and Moodle LMS

Generic Concept	Terminology in IMS-LD	Terminology in Moodle
A subject/course of study	Unit of Learning	Course
A module, unit, project within a course	Unit of Learning	None—perhaps Topic/Week Boxes
A packaged sequence of activities, roles, and files A chain of learning activities and materials	Unit of Learning	None—perhaps Lesson Module, Workshop Module, or the Main Course Centre Column
A task that a learner does	Activity	None (no explicit term, except in proposed

		Project Module)
A learning tool, such as a forum, chat, wiki, quiz	Tool or Service	Activity Module, Activity
Reusable content	Resource, webcontent	Import (Quiz, Glossary, Resource)
A file, link, or external learning activity	None	Resource
A link or process to use external packages	Services	Hotpot, SCORM, QTI
A link to XML external files with IMS-LD global elements	imsldcontent	None
A combination of resources and services	Environment	None—perhaps Course Centre Col.
A specific sequence of activities for some learners	Path	None—perhaps Lesson, Workshop
A requirement to fulfil before playing a UOL	Prerequisite	None
A hidden software that interprets the specification. A rule system	Engine	None
A software or script for designing a learning unit	Editor	“Turn Editing On” Button
A software or script for viewing/operating a learning unit	Player	None—perhaps whole Moodle course with no editing rights
A software to make a course available for view	Publisher	Restore
A summary of a learner’s marks	None—perhaps	Gradebook

	Monitoring Service	
A summary of a learner's participation	Monitoring Service	Activity Report
A summary notation of the IMS-LD	XML manifest	Backup—no files, no user data
A summary of activities and content	None—content not included	Backup—with course files
A summary of activities, content, and user contributions	None—user data not included	Backup--with course files+user data
A course/topic in action	Run of UOL	Course (teachers/student enrolled)
A user-supporting tool, linkable	None	Blocks, Calendar
A variable that is used to build user portfolios	Property	None
A service for user to look at their own properties or others in a structured way	Monitoring Service	None—perhaps Profile, Portfolio
A variable setting of any learning unit to add more personalisation facilities or configure its use	Condition	Activity Module Settings
A message informing of some action, possibly also assigning new learning to roles based on certain events	Notification	Subscriptions (only simple messages)

Table 4. Differences in Terminology for IMS IMS-LD and Moodle LMS

Terminology in IMS-LD	Terminology in Moodle
Run of a Unit of Learning	Course

Unit of Learning	Course export/import file (without runtime data)
Activity-structure of type selection	Topics in a course
Learning activity with one single environment with one tool (depends on the activity)	Activity Module, Activity
Conference of type 'announcement'	Announcement
Conference of type 'asynchronous'	Forum
Conference of type 'synchronous'	Chat
Learning Object of type 'tool'	Wiki
Learning Object of type 'test'	Assessment
Learning Object of type 'tool'	Glossary
Learning Object of type 'tool'	Journal
Learning Object of type 'test'	Quiz

3.4.5. **Pedagogical Descriptiveness**

The IMS-LD specification excels at modeling the structured sequencing of activities/resources and the roles of learners and teachers. A Unit of Learning in IMS-LD is multi-dimensional (Olivier & Tattersall, 2005), including a collection of activities that can be forced-sequenced, conditionally-sequenced, or non-sequenced. Content can be embedded within the Unit of Learning, not just separated in a simple sequence. Currently the Moodle editor has no system creating forced paths of activities, just place holders for separated activities and resources inside a course, which are only visually “connected” in a vertical column of the main interface. Content is also separated as individual files and links, called “resources”. It is one-dimensional in the sense that each resource and activity module (“tool” in IMS-LD) is totally independent and arranged under topic-labels, not formal UOL structures. This is an advantage in terms of ease of design, but a disadvantage when a particular learning unit needs to be containerized and component dependencies described. In setting up a Moodle course, there is a blank column of topics or weeks--almost no structure “out-of-the-box”, but an arbitrarily complex structure can evolve over time. Learners experience maximum control because they can visualise the whole structure and are given full access for free inspection, skipping, jumping back anywhere on the main course page. Teachers, as well, as they edit in Moodle, are often given a start with a set of preformatted choices, but with freedom to reconfigure. This could be called “open learning design”. However, in some IMS-LD editors, such as Reload, a teacher starts with an empty canvas and can decide to design anything, but without the initial prompts to spur/constrain creativity. In this mode of “fixed learning design”, an IMS-LD editor allows the learning designer to decide what parts of a learning flow control are "automated", what parts shall follow hard coded sequencing rules ("conditions", defined by the learning designer), and what parts are just containers for more or less freely negotiated social interactions.

In addition, roles in Moodle are limited to “teachers”, “students”, “course creators”, and “administrators”. Moodle tacitly assumes that the learner's role will remain the same throughout the course. While a learner can be switched to a teacher role in Moodle, only one role can be played at a time and reassignment requires manual intervention by a course instructor. In future implementation plans for Moodle roles (Moodle, 2004b), an unlimited number of definable roles can be created, allowing specific editing and access rights to a defined role. For example, a group leader role might be allowed to edit quizzes, open forums, or assess reports. In an IMS-LD “play”, actors assume roles and sub-roles around the generic types of “learner” and “staff”. Although the IMS-LD specification does not limit editing rights in roles, current IMS-LD tools do not seem to be able to grant editing rights to learner roles. In the IMS-LD specification, roles are more complex, with multiple roles and conditional roles possible. From a teaching perspective, the eventual aim in any learning design tool is to allow instructor/facilitators to assign virtually any non-administrative role to a learner. Learners will become tutors of other learners and need powers to assess, plan, and manage their groups. Pedagogically, many Moodle teachers strive to create a learning environment for students where they get choices (and the freedom to make mistakes). This requires tools that support students with self-monitoring tools (mirrors) covering processes like self-planning, time-management, reflection, re-planning, choice in difficulty level of the activities. IMS-LD must thus allow students to play the design role, giving them editing rights, not just playing rights. A consequence of this pedagogy is that teachers play less of a design role, and more of a facilitator or coach role. It is a complex and heterogeneous process. Complex arrangements cannot be designed without describing and specifying the details and combination of the details of the coach role and the self-coach role.

Finally, the composition of groups within Moodle and IMS-LD are evolving. Student-centred, project-based, and socio-collaborative learning practices place greater emphasis on group-based configurations of learners [104]. The act of group formation may include self-organised, teacher-assigned, or automated assignment according to project interests. Multiple, simultaneous groupings are a necessary requirement as each learning unit has its own collection of groups, each of which may overlap in time. Moodle's group function is for a course-wide, single configuration, useful for defining cohorts that do not change during the term of the course. In the IMS-LD specifications, the group functionality is based on "role-concept". Some IMS-LD tool designers, such as LAMS and elive IMS-LD Suite, found this approach to be less intuitive and extended the current IMS-LD specification on groups. This extension is an open question for future discussion.

3.4.6. **Bricolage**

One of the most striking features of the design approach favored by Moodle is the ease with which course materials can be developed and refined in an iterative fashion. This strategy of course development is very much in keeping with the notion of bricolage [105; 106], and what was earlier called "open learning design". By contrast, the current implementation of Reload with CopperCore distributes a Unit of Learning in a fixed form, unalterable while instruction is in process. A fixed learning design process is useful in some situations, in other situations it may be difficult to adapt the UOL to handle unforeseen circumstances (either emergencies or unanticipated pedagogic opportunities), particularly when they occur after instruction has begun. For example, McAndrew & Weller [107] refer to issues with the "implied prescriptive nature" of the IMS-LD design approach because it seems to conflict with the "flexible and dynamic nature of eLearning". In addition, many instructors who use an LMS to accompany their face-to-face class build their course week-by-week, redesigning the plan both during and after the class. These kinds of "bricoleur-teachers" would prefer to continually capture their learning design after-the-fact and then share these evolving units in a wider community of teachers. Another example of bricolage is this research study, organised as a Moodle course, which began devoid of any design. Taking the IMS-LD book as a "resource" or focus point, we organised discourse around chapters. The content (user postings) and choice of tools were added and rearranged week-by-week (during run-time). This is more than just "filling" a forum during runtime.

The arquitecture approach of Moodle and IMS-LD are quite different. Moodle is based on creation and modification on-the-fly and IMS-LD is based on splitting out design-time from run-time, like all of the eLearning specifications. Moodle allows technically-naïve instructors to create useful learning scenarios almost immediately, and then progressively refine them as their skills improve. This may be a critical factor in Moodle's popularity with teachers. The results of our group's ability to operate IMS-LD tools was mixed. One group member found IMS-LD tools such as the combination of Reload with CopperCore require much more front-loading of skills before useful results can be achieved, and iterative development was inconvenient at best. However, another group member found he was able to produce units using the same tool with 30 minutes of instruction. ASK LDT is another editor that may provide an easy-to-build approach that is focused more on the author's perspective, than the raw specification.

This is an important strategic question for Moodle/IMS-LD integration. If it is likely that the average teacher will be uncomfortable leaving the familiar Moodle environment to author a Unit of Learning with a separate IMS-LD tool, then several other strategies must be considered. On-the-fly creation within Moodle, with a subsequent generation of the IMS-LD specified format via an internal Moodle process seems like a more appealing option. This could be a "template editor" that would support the creation of more course formats that support roles and conditions. Thus for a Moodle course/UOL to be IMS-LD compliant, a way of "capturing" an end state (and stripping the user data) will need to be developed. An improved XML export system for Moodle that supports IMS-LD functionality and specifications may prove to be a not so difficult way to maintain the bricolage design approach.

3.4.7. **Testing IMS-LD tools with Moodle**

There are several IMS-LD related tools currently available: a) engines, b) editors and c) players. In addition, there are more tools in development which combine editing/playing and add GUI-based interfaces. In the following section we comment on how these different kinds of tools could be integrated or used in conjunction with Moodle.

3.4.7.a. ***IMS-LD Engines***

CopperCore [3] is an engine which implements all the levels (A, B, C) of the IMS-LD specification. CopperCore is currently the only IMS-LD engine available and has been extensively tested with a set of examples on Levels A and B [6] and conforms perfectly to the IMS-LD specification. CopperCore provides three APIs: CourseManager, which provides administrative functions (users, runs, roles, publications); IMS-LD Engine, which provides run-time behaviour (activity trees, environment trees, content, completions) and Timer, which provides time-triggered events (various timed completions). CopperCore also provides a library for validating routines and the IMS-LD manifest. It is important to emphasize that CopperCore is an engine, rather than a learning environment or a management system, so it does not provide any user interface for creating IMS-LD packages. In other words, CopperCore was devised to run IMS-LD packages not to design them or edit them.

3.4.7.b. ***IMS-LD Editors***

IMS-LD editors are tools which create UOLs according to the IMS-LD specification. Table 5 provides a list of five examples of IMS-LD editors.

Table 5. Examples of IMS-LD-Compliant Editors

Nr.	Tool Name	Link	Author	Levels
1	CopperAuthor	www.copperauthor.org	OUNL	A
2	Reload IMS-LD Editor	www.reload.ac.uk/ldeditor.html	Reload	A,B,C
3	ASK LDT	www.ask.iti.gr	University of Piraeus	A,B
4	Mot+	www.licef.teluq.uquebec.ca/gp/eng/productions/mot.htm	University of Quebec	A
5	Cosmos	www.unfold-project.net:8085/UNFOLD/general_resources_folder/cosmos_tool.zip	University of Duisburg	A,B,C

Moodle's XML-based backup files can be opened by CopperAuthor and the Reload Editor by modifying the namespace reference, but this is of marginal use, since there is no correspondence between IMS-LD's XML schema and that of Moodle. All these tools are Beta releases and hence they are in a fairly preliminary stage in their development. It is expected that they will reach a higher degree of compliance with IMS-LD in future versions. These tools could easily work alongside with Moodle as external editors of IMS-LD files. With the other applications we have tested, namely ASK LDT, Mot+ and Cosmos, we could not find any way to open Moodle XML-based backup files. Of course, all of these applications can be also used as external editors.

3.4.7.c. ***IMS-LD Players***

There are several IMS-LD players available: CopperCore Player (as a built-in component of the CopperCore engine), Reload IMS-LD Player [5], SLED [42], and Edubox [41]. The CopperCore Player is a working prototype to demonstrate how UOLs run, to check internal functionalities, and to publish instances, roles and users to the engine. It doesn't run outside of its engine and its interface is not very user-friendly. The second player, Reload, has just been updated and offers better support than the previous version for several elements and learning structures of IMS-LD. It still does not implement all the IMS-LD Levels and features but its developers are continuing to work on it and they are confident they will achieve full conformance very soon. As was the case with the CopperCore player, Reload can be used with Moodle as an external web player. SLED is developed under the JISC eLearning Framework. It has delivered an open source player version that integrates services and further development is continuing at the moment. The Edubox player is a full featured EML and IMS-LD player that is used at the OUNL as part of their infrastucture. It can import/export IMS-LD through Educreator but it is not usable for small scale deployment because it can only run on large Unix machines currently.

One of the main goals of this section was to report on a test run involving Moodle and the CopperCore Player in which we tried to determine whether it was possible to integrate both systems in a simple manner. After several attempts, we conclude that Moodle resources can be used inside the CopperCore Player simply by hyperlinking to them. The inverse situation, running the CopperCore Player inside Moodle, is not as convenient. The run of a CopperCore Unit of Learning can be linked as a “Web Link” resource (which causes the browser to navigate away from the Moodle site) but not as a “Web Page” resource (which can embed the resource inside the Moodle user interface). It may be that this problem could be avoided with minor changes minor changes in Moodle's HTML code. Further testing is needed to determine the extent of interoperability.

In general, we are considering at least two levels of integration, a first level with just links between the applications and a second level with two-way communication between the applications. The first level is quite easy to achieve with the current players and engine: a player can be linked from Moodle as an external resource appearing in a new pop-up window or even within the same window. They would be in essence two different applications communicating with each other. Alternatively, the player could be embedded in the Moodle core and executed as an additional module.

In the first scenario, specific linking arrangements have to be made depending on the player but in principle they would not involve any major technical difficulty. The second scenario, that of embedded integration, is considerably more difficult to achieve because of the current state in the development of the players and engine as well as the current stage in the evolution of Moodle. Therefore, the strategy for full integration is still rather uncertain. More extensive testing is needed to find viable ways to allow Moodle to communicate with the existing IMS-LD engine and players. The goal for integration is to allow data to be exchanged between different tools that are running at the same time.

CopperCore has similar data-exchange issues with other applications such as QTI and SCORM engines. Finally, we note that IMS-LD editors cannot be used to edit Moodle courses but, of course, this was not expected to happen until Moodle courses can be exported as IMS-LD Units of Learning.

3.4.7.d. ***IMS-LD-Related GUI-based Editor/Players***

Two drag-and-drop GUI-based editor environments were discussed in this section: LAMS and elive IMS-LD Suite. Elive IMS-LD Suite [108] is not yet available for testing but was described as offering an intuitive GUI-based sequence-editing environment. LAMS has been publicly released and was examined for this report. LAMS (the Learning Activity Management System) is a software system based on the concept of IMS-LD theory which has been in use with teachers and students since mid 2003 [13]. It is an IMS-LD-inspired tool for designing, managing, and delivering online collaborative learning activities. It is important to note that the creators of LAMS do not see this platform as a competing learning management system, but rather as an activity/UOL authoring tool that could be used in conjunction with many LMS. LAMS has an intuitive interface with a visual authoring environment that allows users to create sequences of learning activities with very little effort. Although it is not IMS-LD compliant, LAMS is based on IMS-LD principles and it intends to be IMS-LD Level A compliant by July 2005. The LAMS team has pointed out some problems with IMS-LD that made it difficult for them to implement an intuitive system under specifications. Table 6 shows a summary of the capabilities of all tools mentioned in this section.

Table 6. Roles and Capabilities of IMS-LD and IMS-LD-related Tools

Package Name	IMS-LD editor	Non-IMS-LD editor	Drag/drop editor	IMS-LD Player	Administration
CopperCore Player	X	X	X	O	O
CopperAuthor	O	X	X	X	X
Reload	O	X	X	O	X
ASK LDT	O	X	X	X	X
MOT+	O	X	X	X	X
Cosmos	O	X	X	X	X

LAMS	X	O	O	X	X
Moodle	X	O	X	X	O

At the moment, LAMS is one of the most immediately useful tools for the Moodle community because of its ease-of-use and the willingness of the developers to adapt it into the Moodle environment. Currently, Moodle integrates the exportation facility to LAMS.

Activities/tools in LAMS are similar in function to Moodle activity modules. Moodle 1.5 activity modules include forum, chat, survey, choice, assignment (including journal), resources, grouping, glossary, lesson, wiki, messaging, and optional modules such as book, database, and questionnaire. LAMS 1.0.1's activities are similar, including forum, chat, journal, survey, voting, submit files, share resources, grouping, resource and forum, Q&A+Journal, Voting+Journal, Chat&Subscribe, and Chat&Subscribe+Journal. Moodle has many other activities in development (for example: blog, database, project, document management). The number of activity modules in Moodle is greater than in LAMS, but both sets are capable of building a rich collaborative learning environment. The main difference is that a LAMS activity was built to be "IMS-LD aware", while a Moodle activity is not. With LAMS, you can create a sequence of activities and set the order of activities. Then the created sequence is saved in a private or public repository. If an author needs to modify some aspects, it can be reloaded from the repository and changed. In addition, there is a special kind of activity in LAMS called a Parallel Activity [109] which allows a single person to conduct two streams of activities concurrently on a single screen. We list such activities here: Resource and Forum, Q&A + Journal, Voting + Journal, Chat & Scribe, Chat & Scribe + Journal.

The case of embedded resources was the easiest for Moodle/LAMS interaction. We put LAMS inside Moodle as a resource (a link to a URL). Of course, LAMS and Moodle must have the same session so that we have no login problem. Interoperability interaction was more difficult because Moodle 1.x was designed with no "IMS-LD" framework in mind. Therefore, it is hard for Moodle to interact with any UOL. At the moment, LAMS exports/imports a sequence of learning activities under its own format. Obviously, LAMS cannot use the course data of Moodle and Moodle cannot understand a sequence of LAMS. This, of course, is the reason interoperable specifications such as IMS-LD are needed. Finally, in the case of activities interaction, we found that activities of Moodle and activities of LAMS cannot exchange data or re-use one another because they do not have a common interface for interaction.

In its next version, LAMS will have a common interface to interact with the LAMS core so that third parties can write new tools for LAMS. Moodle tools (activity modules) will probably be able to be used in LAMS (with some small extensions). Finally, the Tools Integration Project (TIP) of JISC has been focusing on integrating some popular open source software (Bodington, LAMS, AMSTOIA) using a WebAuth single sign-on mechanism (Noble, 2005). In this project, LAMS will be able to interact with other systems easily. Moodle should consider adopting this kind of capability.

While these options of external tools may be useful in the short run for integration of IMS-LD into Moodle, a second question is how strategically the Moodle code could integrate IMS-LD internally. Similar to the SCORM integration in Moodle, we could do the following:

a) Create an Export filter that is IMS-LD compatible. This allows the transport of Moodle courses to other IMS-LD compatible players.

b) Create an Import filter that can read the Moodle IMS-LD application profile (so files that are exported with Moodle or created with external tools are compliant with the Moodle application profile)

c) Include an IMS-LD viewer (similar to the SCORM viewer) that can view any imported IMS-LD file that is not conforming to the Moodle application profile.

These are questions that programmers and engineers will need to resolve. Yet from a teacher's point of view, as stated earlier, it would be far preferable to achieve this internal integration, to provide a seemless working environment for a teacher.

3.4.8. **Implications for Future Versions of Moodle**

Moodle has begun an ambitious effort to integrate IMS-LD standards into its future versions. Currently, version 1.7 is not compatible with IMS-LD specifications. However, version 2.0 is intended to provide complete support for the IMS-LD standard, conditional activities, and groups/roles customization at site, course, and activity level. Along with this vision, there are several further implications for Moodle on this pathway: 1) bricoleur tooling, 2) UOL-style authoring, 3) XML code output, 4) roles/conditions/paths, and 5) goals for IMS-LD levels.

3.4.8.a. ***Maintain bricoleur tooling and add an internal UOL editor***

The first implication is for Moodle to maintain its bricoleur mode of design and operation as it incorporates the IMS-LD specification. The French word bricoleur is normally translated as "handyman" or "tinkerer". The pedagogic sense of the word was introduced by Turkle and Papert [106] which grew out of an earlier use by Levi-Strauss [110]. The idea here is that there are two fundamentally different ways of approaching a problem. The "engineer" way involves making careful plans and writing everything down in full detail ahead of time. The "bricoleur" way is more of an organic process of iterative design and refinement. While each approach is useful, the advantage of software designed with bricolage in mind is that the users can start producing useful results immediately. If the software requires lengthy training before worthwhile results can be produced, most teachers will not use it unless forced to do so. Moodle is an excellent example of software designed for bricolage. A naive (or even technophobic) instructor can start doing useful things in Moodle with five minutes of instruction. Seeing an immediate positive result is a powerful motivating factor. There seems however to be no fundamental reason why IMS-LD could not support bricolage by altering the IMS-LD XML tree while the code was running, similar to the way you can use DHTML to alter web pages that have already been loaded (a procedure that tools like CopperCore can support). Consequently, if it is technically possible, we would favor the development of IMS-LD tools that support this work style (preferably internal to Moodle so that an environment familiar to users can be preserved).

3.4.8.b. ***Create UOLs from structured sets of Moodle resources, activities and services***

The Moodle interface is presently organised like a stack of “cards” laid out vertically down the screen. Each card is a square box that represents a week or a topic. A card typically contains a title and some activities and/or resources. Even though a Moodle card is an almost self-contained "piece of learning" and can represent rather complex learning scenarios, it is organized as a rather simple flat structure. The title, activities and resources simply appear one after the other without any other kind of link or internal connection that could provide additional structure or relationships among the different elements in the card. This structure is the most fundamental difference between the central elements in an IMS-LD-specified UOL, and a Moodle “card”. In a UOL, all of its parts are formally related to one another. A UOL typically involves resources and/or services sequenced or linked to each other in some way. In contrast to the flat structure of the Moodle cards, where all activities and resources are visible in the same way for all users, UOLs often involve layers deep of non-visible activities and resources that can be also sequenced or visualized in different ways according to the roles assigned to the different users. In Moodle, as we said, the unit is flat, with no hidden activities behind a title. The title itself is just a label. It cannot hide or pull along any associated parts with it by dragging and dropping.

We propose Moodle add an optional, richer structure to its cards or to the elements within its cards. In other words, incorporate the richer structure of an IMS-LD UOL within a Moodle course but also allowing the option of unstructured elements or components contained in a course. Likewise, it should also be possible to export an entire Moodle course as a UOL. Under this perspective, UOLs would become an additional type of building block in Moodle, next to the traditional flat cards, which the teacher or course designer would have available to construct a wide variety of learning scenarios. The complexity of this kind of design, however, would require a new authoring interface, such as the drag and drop tool developed by LAMS. These movable, swappable cards/units would then be the core objects exchanged in a Moodle repository that is IMS-LD-compliant.

3.4.8.c. ***Generate XML code from Moodle designs after-the-fact***

Moodle needs an 'after-the-fact' tool that builds an XML model after a teacher designs and implements a course. This would 'capture' a model/scenario after or while the learning has taken place. In other words, we would imagine that, as a course makes some progress, the IMS-LD tool analyzes the online patterns and produces an XML model. In addition, a manual editor could then add the face-to-face aspects to the model. Currently in Moodle, there is a basic process happening like this already. Behind the mask of the zip-backup is a non-documented XML-tree. Moodle will need to rework this tree in areas such as automated updating of resources to become fully compatible with IMS-LD. Moodle tends more toward what the authors characterize as 'server-centered' rather than 'manifest-centered', though there are some aspects of Moodle that are reminiscent of a manifest-based approach, in particular the XML format used for backups. It may be that the backup format could migrate toward a more IMS-LD-friendly structure without too much difficulty (perhaps through an XSL transformation). This, however, is a fixed state of a course at one point in time. That is useful for exchange, but of course does not show the changing patterns of a learning design over time, a picture that eventually will prove valuable.

3.4.8.d. ***Add multiple, definable, conditional roles***

Moodle needs to implement definable roles as outlined in the Implementation Plan for Roles, and should move towards the capability to incorporate multiple roles, conditional roles and temporary roles. One goal is to create an intermediary role between teacher and student—such as "tutor" with limited teaching permissions. We can also define roles such as mentor and mentee. These roles would then be defined at the site level, course level, and activity level, possibly allowing multiple roles within the same course. However, it appears that the IMS-LD concept can go further with "multiple" roles. We assume this means that someone could have several simultaneous roles in a course. For example, within the project area B, John is tutor, but in project area C, he is a novice student. This would be very appealing to instructors who would very much like to take a single activity and assign all of our students to be "teachers" in that activity alone. For example, teachers often ask students to create quizzes on paper, and then assign one to a teacher role to input the questions into Moodle. However, this would work more smoothly if it could become a configuration option inside Moodle. Another concept is conditional roles. A student would automatically be given a different role when certain conditions are triggered. This operation is much like moving up to the next level in a game. To do this, the user tables may need extra fields to store temporary role flags (during a course) or even longitudinal flags (preferred learning style), and even the combination of these flags. That process could be easy, but the difficulty would be implementing the engine that evaluates a script against these roles.

*3.4.8.e. **Aim for IMS-LD Levels A, B, C***

Currently, Level A export funcionality is under development to be delivered in 2007. As Moodle doesn't allow yet the features of Levels B and C (like properties, conditions or notifications) it is not appropriate to suggest their direct implementation. Nevertheless, at least two points should be considered regarding IMS-LD levels. First, IMS-LD levels are a distinction for implementers, not users. They are levels of the effort to implement the related functionality, not levels of the complexity of the learning designs that are created with a tool. This can result in situations where one has rather simple learning scenarios (from a teachers point of view), but these cannot be implemented on Level A, because, for instance, certain properties are required. Second, when someone decides to start with a Level A implementation, this should be done with Level B and C "in mind". The implementation of a sequencing mechanism in terms of "acts", for instance, will vary considerably depending on whether we plan to extend it in the future with sequencing triggered by properties and conditions. There is also a limit as to how much complexity can be reduced when the views and needs of the different stakeholders are considered. For this reason, implementation of all the three levels should be our goal from the outset.

3.4.9. **Implications for IMS-LD: future versions**

Creating the universal learning design protocol, IMS-LD, like any evolving standard, is in a continual process of development. Teacher practitioners such as those in the Moodle community are eager to contribute to this development because of their enthusiasm to begin sharing materials and designs in an inter-LMS exchange system. In this section, we shall outline some implications for the further development of IMS-LD from this teacher-developer perspective as IMS-LD moves to become more widely accepted as a language of exchange. We will separate our recommendations and implications on three levels, beginning with the concept or theory of IMS-LD, then move to the specifications of IMS-LD, and finally the implementation of IMS-LD tools. It is perhaps overly audacious on our part to suggest changes after only a few months of immersion, so we ask IMS-LD developers to accept our apologies for any incorrect assumptions or immature understandings as we try to grapple with the intentions and concretions of IMS-LD.

3.4.9.a. ***Maintain current conceptual framework***

The conceptual framework of IMS-LD is powerful and appears to hold the core requirements that users of Moodle value and require. It goes beyond single-learner-in-isolation standards, such as SCORM, to include collaborative modes of learning with flexible roles. The core principles or requirements are all in alignment with the principles that Moodle users would generally agree on. These eight principles defined by Koper (2005, p. 19) can be summarised as:

- IMS-LD must be comprehensive: including objects, services, activities, roles, solitary/group models.
- IMS-LD must support blended learning: face-to-face integration as well as pure online learning.
- IMS-LD must be flexible: supporting all theories of learning, pedagogically neutral.
- IMS-LD must describe conditions of learning: tailoring the design to specific learners or situations.
- IMS-LD must stimulate reuse: portability, arrange-ability, addition/subtraction of parts.
- IMS-LD must be standardised: operate with other standard notations (i.e.: IMS-QTI)
- IMS-LD must be automatised: provide a language for automatic processing
- IMS-LD must be abstracted: for repeated execution in different settings and people.

3.4.9.b. ***Allow pluralistic design philosophies in IMS-LD tools***

While these core requirements provide an excellent framework for exchange of learning, questions have been raised as to the design methodology of specific IMS-LD tools. In other words, while the IMS-LD specification aims to be pedagogically-neutral, the IMS-LD-tools may prescribe a particular design methodology. Implicit in the design of any learning activity is an epistemological question about the nature of design. The nature of design has been classically conceived in a "pre-engineer and run" paradigm. Diffusion models of innovation (Rogers, 2003) operate in a similar way. First, an innovator constructs a new design, and then the design is disseminated. In contrast to this, there is a translation/transformation model of innovation in which designs are co-created by environments and actors in a way that continually transforms the network of actions

[111]. The properties of the design itself are actually less important than the reconfigured network of actions and the very process by which this network of actions and relationships is reconfigured in a learning community. This community-based, ecological paradigm of learning may be a theoretical concern that IMS-LD will need to wrestle with. The most common concern that facilitators in the Moodle community expressed was the pre-engineering mode of operation that they felt they were being forced into when working with IMS-LD. Moodle itself offers three pre-engineered formats (topic format, social format, and weekly format, yet within the topic and weekly format it not necessary to pre-design any aspect of the course. In addition, formats in Moodle are pluggable, with new formats in development such the Project Format and the Sequenced-Activity Format (LAMS).

The design-on-the-fly ability of the Moodle LMS was a critical attribute that no one was willing to part with. One commenter said, "freedom from design is just as important as freedom in design". In other words, it might be productive to distinguish between different types of 'design'--a conscious/explicit process of design and an unconscious/non-explicit mode of designing and compare IMS-LD tools through those criteria. The ability to design unconsciously is an inherent and useful practice that is embedded in the daily routine of teaching. In some ways, Moodle emulates this non-explicit design. The ability of IMS-LD tools to offer similar freedom may have to do with their design philosophy or current stage of development.

3.4.9.c. ***Extend IMS-LD Specifications in services for Collaborative Learning***

Another question was raised about whether IMS-LD was sufficiently developed to handle all the social dimensions of learning in Moodle. For example, we noticed that two IMS-LD-related tools, LAMS and elive IMS-LD-Suite, had found it necessary to extend the specifications of IMS-LD to handle the complexity of groups in learning. These two tools use runtime extensions to manage group functions in order to work on collaborative learning. Besides, although it is not its final goal, IMS-LD could include some components or an extension of the specification to incorporate services, such as fora or online events, for instance. At the moment each tool implementer is free to chose their own implementation. This is only a standardisation issue when we want to exchange these specific implementations. IMS is not working on these additional specifications because some tool developers argue that exchange is not necessary. However, in other ways IMS-LD appeared to be very ambitious in some aspects of social learning. In Halm, Olivier, Farooq & Hoadley [112] we found the peer-to-peer model of collaborative learning to be beyond the boundaries of our current thought. If IMS-LD can accommodate that decentralised kind of learning, it should have little problem with the issues surrounding group organisation and operation. In addition, IMS-LD seems not to have a specific way to handle forums, but just makes a reference to them, perhaps so the IMS-LD package itself is not tied to any specific forum setup. Moodle allows a number of definable properties to forums, and the varieties of group process produced by these configurable rules can and should be modeled. In addition, Moodle has numerous ways of handling unstructured communication, not just for discourse (wiki, blogs, instant messenger), but also for structured data (glossary, blocks, database). New code may need to be written in IMS-LD players to make them operate smoothly with any forum-oriented LMS such as Moodle. In the JISC ELF framework, this is solved by using web services whenever a tool cannot connect with a run of a UOL.

3.4.9.d. ***Clarify Administrative and Learning Services***

The general purpose of IMS-LD is not to provide a set of services because IMS-LD is not a LMS. Moodle has a rich set of student-monitoring services such as Gradebook, Activity Reports, Block Reports, Logs, and Portfolios that are an essential part of the learning environment. Bearing in mind the current state of development of on-line learning environments, it is not an exaggeration to say that the usefulness of most UOLs will depend more and more on the appropriate integration and configuration of these types of components. While IMS-LD specifications are capable of modeling Units of Learning it would be needed to integrate both, administrative and learning services, in order to provide teachers and learning users of useful supporting tools. For example, Moodle's tight integration of activity modules and activity reports and the effect this has on teacher coaching of students demonstrates that many "administrative services" in the learning environment have an impact on the success of learning.

3.4.9.e. ***Promote GUI-based IMS-LD Tools***

The number of IMS-LD related tools is growing rapidly. For Moodle users, LAMS is one of the most intuitive tools because teachers can create a sequence of learning activities by dragging and dropping. Most other tools (i.e. CopperAuthor, Reload) are designed for users who are familiar with IMS-LD concepts (play, act, role-part, etc) and may be more suitable for developers and designers than for the average teacher. IMS-LD tools should be more intuitive and easy-to-use so that non-technicians can use them to create and exchange UOLs.

3.5 . **An example of integration: mapping IMS Learning Design and Moodle**

The specification IMS Learning Design (IMS-LD) and Moodle look for a common understanding focused on the integration of both eLearning approches. The final goal is that Moodle will be able to play an IMS-LD package and any IMS-LD tool will be able to import a Moodle course and use it as a base, or even to import a Unit of Learning made in IMS-LD to be used and played in the Moodle Course Management System. The Unit of Learning in IMS-LD (UoL) and the course in Moodle become the perfect marriage where to find several elements that should match one to each other. This section show how to make this understanding between notations, mapping related elements in both to get a list of pairs easy to translate from one to each other, and to define also a list of requirements for this protocol [113].

3.5.1. **Introduction: why mapping IMS-LD and Moodle**

IMS Learning Design [114] is aimed to transform regular lesson plans into interoperable Units of Learning (UoL). This specificatiobn is able to use any pedagogical model to get a UoL runnable and editable in an interopable way. Furthermore, IMS-LD is able to describe a full learning flow with several elements -such as roles, activities, environments or resources- and features -such as properties, conditions, monitoring services or notifications- [115; 116]. The interoperable UoL can be edited with any IMS-LD editor, like CopperAuthor [117] or Reload IMS-LD Editor [118], and later played with any IMS-LD player, like CopperCore Player [119], Reload IMS-LD Player [118] and Sled [42]. On the other hand, there are a couple of challenges related to the spec: a) IMD IMS-LD has no Learning Management System able to handle and to play these UoLs; and b) there is a low-level approach of the editors and the technical profile that a learning designer or teacher needs to hold in order to create a UoL in an easy way [120].

Moodle [10] is a Course Management System (CMS) easy to install and to use and wide abroad disseminated, with more than 100.000 registered users, 12.000 registered websites and translation into 70 languages. Also, Moodle has a very strong virtual community of active users carrying out an increasing amount of face-to-face and online activities, and supporting each other via the official site[3] and a number of ad hoc assorted websites. Moodle is able to manage every feature of a course and the related environment, such as user definition, groups, access, resources, internal links and a long et cetera. This is a difference with IMS-LD. From a social constructionist approach a course is created from scratch in a few minutes. Moodle also allows for the execution of other information packages, like Scorm and Lams, as an encapsulated module inside a course. On the other hand, the pedagogical expressiveness of Moodle is limited by the absence of features included in the IMS-LD model, such as defined adaptive learning flow, flexible roles and adaptive content. Currently, Moodle is working intensively to provide these missing features so that it is able to support IMS-LD more fully over time.

[3] http://moodle.org

The main issue is that IMS-LD and Moodle are not comparable at all, as they have different approaches to work on eLearning as well as they are different releases [100].

In this context, the mutual understanding between IMS-LD and Moodle seems like an improvement for both parts. IMS-LD provides a pedagogical flexible approach in the creation of UoLs, as well as the specification support and the technical background focused on standardization and interoperability, and Moodle provides a well-known and easy-to-learn CMS and an active community of non-technical profile. To this end, Moodle and The Open University of The Netherlands[4] founded a working group in June 2005, then hosted by the UNFOLD Project[5] and Learning Network for Learning Design[6] and currently supported by the ProLearn[7] and TENCompetence[8] projects.

3.5.2. How to make it: basic structure and mapping

There are a few attempts to integrate UoLs with stand-alone IMS-LD players like Sled and Moodle [62]. Sled is able to run a UoL stored in an IMS-LD server via an Internet Explorer player in the client (Telcert). A link from a Moodle resource to the ip address where Sled is allocated allows a simple first level of integration. This structure is not focused on the reusability of the lesson plans but on the integration of current systems to form a de facto more complex approach, collecting several technologies.

What the mapping of Moodle and IMS-LD aims to is the re-usability of a lesson plan/course/UoL of one of them into the another, to be used as a base for a further development or as they are actually defined. Furthermore, it is focused on the interoperability and the reusability of an information package/UoL, no matter the original platform that is used for it.

In order to achieve the best understanding between IMS-LD and Moodle, the mapping process is divided in three steps:

1. Moodle is able to export one course to a UoL, translating the Moodle notation to IMS-LD

2. Moodle is able to import one UoL into the Content Management System and translate the IMS-LD notation into a Moodle notation

3. Moodle is able to play a UoL inside the system, following one of the approaches on types of integration between information packages and players suggested in Tattersall et al, 2006: Moodle stores an IMS-LD information package and it runs an internal player

[4] www.ou.nl and www.learningnetworks.org
[5] www.unfold-project.net
[6] http://imsld.learningnetworks.org
[7] www.prolearn-project.org
[8] www.tencompetence.org

To realize this three blocks we need to establish a general framework:

- Besides the Moodle course, the rest of the Course Management System environment is out of scope (calendars, blocks, log-in, language...) as they are used as processes and instructions and not like a core part of the basic unit of interchange for information (i.e., a Moodle course or an IMS-LD UoL)
- There is a need of matching every single Moodle feature-component to an equivalent in IMS-LD or to define it like an external process/instruction
- In order to make a taxonomy of the elements in a Moodle course and to find a mirror in the IMS-LD specification, we define four main groups: 1) Setting -basic configuration-, 2) Activity, 3) Resource, 4) Administration –out of scope-

If we take a Moodle course we can match every element of the course with an element in the basic structure of an IMS-LD Unit of Learning. Table 7 shows the result:

Table 7. Elements in an IMS-LD Unit of Learning mapped to elements in a Moodle course

Issue	Moodle feature-component	IMS-LD structure	Remarks
0	Full course	1 UoL, 1 play, 1 act, 1 activity structure (type selection)	
1.1	Setting: Full name	Title of Learning Design	
1.2	Setting: Short name	Title of Play	
1.3	Setting: Hidden sections	Activity Structure:IsVisible	Possible, but suggested for a next iteration
1.4	Setting: summary	IMS-LD learning objectives	
1.5	Setting: Your word for Students	roles:learner:title	
1.6	Setting: Your word for Teachers	roles:staff:title	
2.0	Topic	Learning activity	If a resource or an activity is defined there will be an environment If there is no activity description in Topic it should be auto-generated with a standard text or just empty
2.1	Topic 0	Learning activity (first one)	
2.2	Summary of Topic	Activity Description	
2.3	Activity: Chat	Synchronous conference (conference-type)	Services
2.4	Activity: Forum	Asynchronous conference (conference-type)	Services
2.5	Activity: Glossary	-	Generate as dynamic HTML
2.6	Activity: Quiz	-	Possibly it generates QTI
2.7	Activity: Scorm	No match	For a next iteration
2.8	Activity: Survey	External	Generated by Level B
2.9	Activity: Workshop	External	Generated by Level B
3.0	Resource	Learning object	All the resources ended as learning objects except Directory and Link
3.1	Resource: Directory	-	Generated by Level B
3.2	Resource: Label	Learning object	Could map to a title of environment if needed
3.3	Resource: Link to file	Learning object	File should be included in content pack
3.4	Resource: Link to website	Learning object	Link should be an absolute URI

3.5.3. The first step: exporting a Moodle course to an IMS-LD Unit of Learning

The first step in the integration process is focused on the exportation of a Moodle single course to an IMS-LD UoL. In order to achieve this goal we establish a list of assumptions: a) there is no round tripping for the first stage, b) the exportation is completely made in batch mode (therefore, no dialog nor user interaction), and c) this task is planned as an iterative process where the first iteration gets the basic skeleton for conversion and the subsequent versions will extend it.

IMS-LD is defined as a metaphor built around a theater using roles, plays and acts. Inside, some elements describe the educational framework: learning objectives, activities, environments, property of visibility, method, type of learning flow, etc. In Figure 27, there are several couples of elements IMS-LD-Moodle for the most basic structure of a Moodle course:

IMS Learning Design UoL	Moodle course	
Unit of learning	0	1 Uol, 1 play, 1 act, 1 AS (type selection)
Title	1.1	Full name
Learning objectives	1.4	Summary
Roles	1.5	Students
	1.6	Teachers
Play		
Title	1.2	Short name
Act		
Title		
Activity structure		
Title		
isVisible	1.3	Hidden sections
Learning Activity	2.0, 2.1	Learning Activity
item-description	2.2	Summary of topic
Learning Objectives		
Environment	Several	Activity and resource: External, Learning Object or Conference
isVisible	2.14	Activity: Assignment: visible/hidden
	3.6	Resource: visible/hidden

Figure 27. Basic match between an IMS-LD UoL and a Moodle course

3.5.4. Mapping services

Some activities in Moodle need a special process, as they have some basic data used for the appropiate execution (i.e., forum, wiki, quiz…). Every activity or resource in Moodle needs to export some additional information that is not supported by IMS-LD, i.e., timing in Forums or grades in Workshops. If IMS-LD is not able to manage this information it will be lost and no later retrieval will be possible from IMS-LD to Moodle, in future versions, although no round tripping is assumed in the first approach.

A possible way is to associate a file with all the extra information related to an activity. For instance, a Moodle Forum is matched to an asynchronous conference-type service in IMS-LD and there is a linked file with the information about starting time, ending time, or discussions policy. We could call the file serviceparams.xml. This file is a resource in the content package, type servicecontent, although other types are possible: webcontent, imsldcontent and imsqti_item_xmlv2p0. The field serviceparams.xml needs to be associated with the service service-conference, but this is not possible yet in the current IMS-LD 1.0´s information model. One approach is to associate an additional learning-object with a service. The final approach is as follows:

```
<environment>
     identifier="env-Topic-0-News-Forum">
      <title> Moodle Summary Topic </title>
         <service identifier="service-conference" isvisible="true">
           <conference conference-type="synchronous">
            ...
        </conference>
      </service>
      <learning-object identifier="ForumParams">
          <item identifier="PARAMS" identifierref="RefToParams"/>
      </learning-object>
</environment>

…

<imscp:resource identifier="RefToParams" type="servicecontent"
href="params.xml">
      <imscp:file href="params.xml" />
</imscp:resource>
```

Following this structure we could map any resource or activity without any lost of important information in Moodle. Although this approach of Moodle is mis-using the notion of a learning object, it could serve as a temporary solution until a modification of the XML Schema in a new version of IMS-LD could happen.

3.6 . Review of IMS-LD based on the integration with the LMS Moodle

The previous section has been a review by teacher-developers of the complexities of integrating the Learning Design concepts, specification, and tools with an open source LMS, Moodle. We have attempted to view the two from an outsider perspective, though as Moodle users, it is not always possible for us to avoid certain presumptions. The first subsection compared Moodle and Learning Design in its terminology and pedagogical descriptiveness, and the contrast between bricoleur and pre-engineered design. The following subsection examined current IMS-LD tools and found that the distance for integration was far closer than we imagined. The LAMS/Moodle integration was an encouraging step towards IMS-LD compliance.

In the fourth subsection, implications for Moodle were outlined. The Moodle community needs to consciously preserve its intuitive structure for designing courses. A post-run capturing of IMS-LD-based XML schema will be needed to achieve both IMS-LD compliance and bricoleur design. Multiple roles in a structured Unit of Learning, with conditions and paths, are future requirements. Levels A, B, and C of the IMS-LD specifications should be the goal with an eventual internal Moodle editor for creating IMS-LD UOL.

In the fifth subsection, our group suggested implications for the further development of the IMS-LD specification and tools. We found the current conceptual framework to be comprehensive and very appropriate for modeling education in Moodle. With respect to the specification, we expressed concern that bricoleur-style design philosophies, collaborative learning complexities, and comprehensive learning services be well accommodated. Finally, in IMS-LD tools, we would support concurrent development of IMS-LD repositories to provide support for the exchange of UOL. In addition, it is important to promote intuitive design environments that are teacher and learner-friendly. These implications are summarised in Table 8.

Table 8. Implications for Moodle and IMS-LD Integration

Implications for Moodle	Implications for IMS-LD
1. Maintain bricoler tooling & add internal UOL editor	1. Maintain current conceptual framework
2. Create UOLs from structured sets of Moodle resources, activities and services	2. Allow pluralistic design philosophies in IMS-LD tools
3. Generate XML code from Moodle designs after-the-fact	3. Extend IMS-LD Specifications in collaborative learning
4. Add multiple, definable, conditional roles	4. Clarify administrative and learning services
5. Aim for IMS-LD Levels A, B, and C	5. Promote GUI-based IMS-LD Tools

The process of integrating the IMS-LD specification into Moodle is happening at a far higher pace than anticipated when we began this study, in 2005. Step-by-step integration initiatives are already underway. With a distributed, online learning community, teacher-developers from secondary and tertiary institutions were able to grasp much of the complex conceptual framework of Learning Design and dialogue on the issues. This framework is now undergoing a translation into practice and will continue to be transformed as teachers and learners take fuller ownership of IMS-LD.

IMS Learning Design and Moodle are two entities that work on the process to reach a kind of integration that allows for the exportation of a Moodle course to an IMS-LD format, the importation a UoL to a Moodle CMS, and the execution of an IMS-LD information package into a Moodle CMS. Following this three-step process a working group formed by Moodle and The Open University of The Netherlands looks for an understanding between both eLearning notations in order to achieve some degree of interoperability and re-usability of online lesson plans and courses.

The first step to take is focused on the exportation issue and it aims to get a basic mapping between both, taking a simple Moodle course that could be extended with additional features in a second round. For this exportation, a challenge is to keep some information in services or activities that Moodle uses for their configuration and execution. An approach points to keep this information in a file stored inside the information package as a resource and to link it as a learning object to the called service. This way, all the needed information is fully exported to an IMS-LD package but leaving it out of the main manifest, where it couldn´t be properly handle, as there are no possible match yet. A second approach points to a modification of the XML Schema in IMS-LD v1.0 where this situation could be manage directly.

3.7 . Integration and Learning Objects: IMS-LD and eGames as internally modelled resources

IMS-LD supports the expression of a wide variety of pedagogical models. If a lesson plan is taken from a general or specific data base (like Merlot, www.merlot.org) it would be able to be modelled with IMS-LD.

In adition, during the development of IMS-LD it has not been tested extensively whether it could model educational games, although it has several features to suggest that it can be used to do it.

To what extend, can be IMS Learning Design be used to model educational games?. This question will provide some answers to general questions coming from a broad study of a concrete field, concerning Learning Objects and the integration with IMS-LD. When IMS-LD can model eGames the full integration of this learning object is an improvement to the pedagogical expresiveness of the specification. When IMS-LD cannot make it, the level of integration of eGames as learning objects with the specification should be defined. IMS-LD's generality implies that it should also be capable of modeling educational games. Issues on synchronization, grouping, multi-role playing, timing, collaborative working, dynamic feedback, adaptive learning and more are implicit or explicit in educational gaming and can be approached with it. Of course, some framework must be defined to describe what is considered as an educational game and when. Lots of games, commercial and free, are shown as educational ones and lots of them don't run under educational requirements. However, both types could be used or not as educational games depending on the frame, the objectives, the context and the moment they are run. We will establish a classification of educational games that we will used later on to define which ones are affordable with IMS-LD and which ones are not.

In this section this question will be address and solved analysing the current research on the topic, dissecting the different kinds of educational games and its key features and proposing ways to model them with IMS-LD. First we will analyse the variety of educational games that are described in literature. Then we will categorize their major features (these are the requirements for IMS-LD). Afterwards, we will discuss the general structure of IMS-LD and will look which of the general game features can be modelled in IMS-LD and which not (or not directly but via a workaround).

Following this process we will identify: a) what aspects of games can be modelled and which not, b) looking at the variety of games available we will identify which type of games can be modelled with IMS-LD and which types can't, and c) suggest ways of implementation for the major challenges.

In the end, a full game consisting of several stand-alone modules but related each other will be described as an example developed in IMS-LD. Its aim is to provide support to students of first grade of primary school (basically based on language acquisition and fixing).

3.7.1. Background on educational electronic games and simulations (eGames)

EGames have become an important topic in the recent, and not so recent history of education. Gaming itself is becoming a key issue in education and has been widely researched in the last 40 years [121; 122]. With the first appearance of the computers at schools in mid 80´s, gaming was supported by computer applications that tried to look for an integration between pedagogical principles and computer science facilities [123-125]. In the mid 90´s the Internet started to provide new perspectives for eGames. A range of new possibilities was available, such as collaborative worldwide extended multi-player sessions, instant messaging, instant updating of settings and multi-language support. The array of features it is still growing, and is attractive for regular users, but also for learners and teachers [126; 127].

Gaming and learning can become a perfect marriage. In order to achieve educational goals, several interactive learning techniques can be used in, or in connection with games. Some examples are learning from mistakes, goal-oriented learning, role playing and constructivist learning [127]. Games allow players to experience, to try, to improve skills, to learn content and to practice strategy (Turkle, 1995; Piaget, 1962; Vigotsky, 1978). When playing games, one of the most commonly applied strategies is trial-and-error. Trial-and-error behaviour of a learner during playing is characterized by the absence of a systematic strategy [128], although at the same time it is a primary way to learn and to keep the player motivated [129]. The mentioned learning techniques can be implemented in games, so that the game itself becomes fully integrated in the learning process, instead of remaining as an isolated stand-alone resource [130]. In doing so, any generic game can be an educational game if fully integrated in a learning process [131]. Furthermore, teachers are able to make good use of generic games, educational games and simulations in their daily teaching activities [132-135]. However, a better development of an integrated model is needed in order to achieve learning goals better and to fully take advantage of the power and potential of games in education [136].

EGames are attractive, addictive, fashionable and elicit emotional reactions in players, such as wonder, the feeling of power, or even aggression [137]. These features make it interesting to attempt to engage learners in a gaming activity. Engagement and educational goals can mutually support each other in the same environment to achieve specific targets. But they can also support rather accurate episodes of history [138], real systems [139], complex popular events [140] or board games [141], just to mention a few. With eGames, the users can also learn content [142], research in human relationships [143], improve personal and social skills [144] and work on strategies [145-148]. This last category, strategy, is one of the bases of the case study presented in Section 3.7.5.

Games have the power of engaging people. They are fun and provide interaction, interactivity, problem solving, story and other elements that give the user involvement, structure, motivation and creativity, among other benefits (Prensky, 2001). They also provide one or several focused goals, well defined rules, challenging and clear tasks, affiliation, choice and the right to be wrong without adverse consequences (Jones et al, 1994). A key factor is that games also provide outcomes and feedback in real-time (Rieber, 1996; Laurillard, 2002), guiding the user in taking actions, helping to focus activity and decisions and the evolution of the story. They are attractive for the players but also for the teachers as they engage and excite the students as well as provide a mean of interaction and learning.

3.7.2. **What is an educational electronic game. Features**

Although is not the main drive of this section to dig into a very detailed taxonomy of games, it does to approach the problems and challenges generated by them and how to solve them inside IMS-LD. Thus, it is necessary to draw a basic profile to be used as a conceptual framework for this paper and for a further research. So, an enumeration of features needed in a game to be considered as an educational electronic game will be analysed in cascade. This means, to define first the general features that a game must have to be called this way, secondly, what components should be added to use it as an educative one and, thirdly, what needs this game has to meet to take it as electronic.

Reading the classical descriptions on the topic (Huizinga, 1971 y Callois, 1967) some features are exhaustively described stressing what a game must accomplished:

- It is a free action, begun and finished by the user as he wants. It is also
- Imaginary, parallel to the real world, replicating an universe or an activity without any consequence on actual issues
- Limited, in time and space
- Following a set of rules, an specific and private framework
- With an uncertain solution and development, due to every run, every play, is different and depends of unpredictable behaviours of the users

- If a game needs to be an educational one, some additional features must complement this enumeration (Sutton-Smith, 2001 y Salen, 2003):
- It has to start with a premise to be solved
- Being unproductive, it doesn't generate any property or wealthy. The last drive is the gaming activity itself
- With at least one right solution always
- With something to be learnt by the user/player, meaning introducing new knowledge, fixing previous acquired knowledge, training skills, sharing experiences, discovering new concepts, developing outcomes

Last, to be an electronic game (Wolf, 2003) the educational game must be run in any electronic platform, such as a computer, an online terminal, a video-player, a pda, a mobile phone and etcetera.

When matching these three steps a game should be considered as an educational electronic one, although not all of the features have to be fulfilled.

Taxonomy of educational electronic games

Before starting a limited taxonomy and, therefore, a restricted and uncompleted categorization, of this kind of games, it is needed to be mentioned the playful service and the didactical basement of an activity itself. Every game and every activity can incorporate an educational element defining the related scenario that seeks educational objectives. Thus, and depending on the final goal, the original drive and the environment, any game can be considered educational, beyond its own definition. The focus of this paper, though, is on the really specific educational games described as is, and that accomplish the rules described above.

Bearing this qualification in mind and following the intrinsic essence of an educational game, there are several kinds of them: chance, action, builders/creators, board, strategy, sports, management/planning, intelligence, platforms, replicates of scenarios or universes and simulators. Most of the times, this classification doesn't follow pure definitions and it is usual to find mixed models joining together several features, as simulators and action games, universe replicates and builders or strategy and platforms.

3.7.3. Main components of an educational electronic game

A broad variety of games has been taken as a sample, following the current market of sellers, producers and critics. Several catalogues and magazines have been used as a base (www.amazon.com, http://cgw.1up.com, www.gamers.com, www.pcgamer.com, http://buy.soft32.com) matching their content with the taxonomy described previously.

Among all of them, the main components have been extracted, splitting the final list in two parts, technical components and didactical components (see Table 9):

Table 9. Main components of educational electronic games

Components	Remarks
Didactical components	
Single or multiple solution	There is only one chance or several feasible solutions to complete the game satisfactorily
Opened or closed solution	The solution is internally chosen as a result of a selection between several possible solutions or the user can generate his own solution not previously stored inside the application
Individual or collaborative solution	One or several players are needed for a happy result
Collaborative or competitive execution	The solution must be reached as a result of a collaborative work or a competition is established to get it
With dynamic feedback	A valuation of the user action is provided and it is conditioning the flow in the game
With adaptive learning	Personalizing contents, itineraries, assessments, depending on the user profile, on his rhythm and on his playing
With incremental or isolated learning	There is a number of progressive learning levels or they are isolated independent cells
Technical components	
Single or multiple player	Just one single user at the same time or

	several users can be playing
With or without users grouping	Sharing resources, goals and communication
Local or distributed running	A connection to a network or to Internet is needed for a multi-terminal/multi-user execution
Synchronous or asynchronous running	Communication at real time of broadcasting
With or without multimedia resources (video/audio)	Multimedia files are integrated and used, such as video or audio files
With strong graphical support	Graphical illustration is a bottleneck for responding time and storing capacity because of the amount of information
With 2D or 3D graphical design	Illustration comes from two dimensional and flat sources or from three dimensions and incorporates depth as a key factor. This means more resources consumption
With vector or bitmap images	Illustration is math-scalable-based without deformation or it has a photographic quality
With a run time engine or with engine not adaptable at run time	There is an AI engine that make calculations at the run time and modify structures and the game itself or everything is defined at design time
With editable or static fields	Fields can change their content during the play
With personalised features or static profiles	Specific features of the play and the user can be modified during the running
Saving and reading external files	Retrieving and storing some user, play or behaviour information, for instance

With communication from and to other applications/tools	Sending information and interchanging variables during the play
With or without interaction in the real world. Blended-gaming or b-gaming	It is an electronic stand-alone game or an interaction with the real world is needed

It is important to enhance, reading both, the technical and the didactical components, the usual combination between them. Almost none of them can be selected as a stand-alone feature, always supporting others and combining sources and outcomes. Therefore, the implementation in IMS-LD of any of them entails the nested joint implementation of a set of them. For instance, technical components as “Local or distributed running” comes with “users grouping”; also the didactical components “dynamic feedback” goes with “adaptive learning”. Likewise, and merging both lists, the didactical component “incremental learning” comes with the technical one “saving and loading external files” and the didactical one “collaborative solving” relates to the technical component “single or multiple player”.

3.7.4. **What components can be modelled with IMS-LD**

Taking into account the previous list of technical and didactical components a correspondence can be established between them and their chance to be modelled with IMS-LD. The next table (see Table 10) has four columns: 1) analysed component, 2) whether it can be modelled with IMS-LD or not, 3) what level of the specification is required (just a reminder, there are three levels, A, B and C) and 4) any additional remark if it is needed.

Table 10. Main components of the games to be modelled in IMS-LD

Components	IMS-LD	Level	Remark
Didactical components			
Single or multiple solution	Yes	A	
Opened or closed solution	Yes	A	
Individual or collaborative solution	Yes	B	XML(1)
Collaborative or competitive execution	Yes	B	XML
With dynamic feedback	Yes	B	XML
With adaptive learning	Yes	B	XML

With incremental or isolated learning	Yes	B	XML
Single or multiple solution	Yes	A	
Technical components			
Single or multiple player	Yes	B	XML
With or without users grouping	Yes	A	
Local or distributed running	Yes	A	
Synchronous or asynchronous running	Yes	A	
With or without multimedia resources (video/audio)	Yes	A	
With strong graphical support	Yes	A	
With 2D or 3D graphical design	Yes	A	
With vector or bitmap images	Yes	A	
With a run time engine or with engine not adaptable at run time	No	-	Depends on the executable module itself, and not on IMS-LD
With editable or static fields	Yes	B	XML
With personalised features or static profiles	Yes	B	XML
Saving and reading external files	No	-	Depends on the executable module itself, and not on IMS-LD

With communication from and to other applications/tools	Yes	B	XML
With or without interaction in the real world. Blended-gaming or b-gaming	Yes	A	
(1) *Additional XML needed component*			

The first read of Table 10 shows that, all the components, both didactical and technical, can be modelled in IMS-LD. Concerning Level A there are two reasons: a) IMS-LD has integrated the feature itself: grouping, single or multiple solution, opened or closed solution, synchronous or asynchronous solution; and b) the feature doesn't depend directly on IMS-LD but can be supported properly: 2D/3D design, strong graphical support, multimedia resources. Concerning Level B, also the same two reasons with different arguments: a) IMS-LD has built-in the component or can be programmed with some additional help of global elements, properties, conditions and monitoring services in XML, like editable fields, personalised features, individual or multiple running, collaborative or competitive running; and b) it doesn't depends on IMS-LD but it is supported: communication from and to other applications.

3.7.5. **Three models of implementation for an example Unit of Learning**

To show an example, a Unit of Learning has been developed to support the storing and retrieving information process in a external file inside IMS-LD and the use of its content if the learning flow (Figure 28). The chosen game to support our argument is called Caminatas and it was included for the first time in a multi-rom providing support for language acquisition and fixing in Primary School [142]. Furtheron, we will use the same UoL to explained other integration model.

Caminatas consists of several stand-alone modules with basic data communication, grouped around a central main module to grant access and share tasks. It has some features to be personalised (user name), some setting-up for the full multi-rom (ambient audio) and some adaptive configuration (visibility of intros just in the first running of every instance). It was scripted and programmed in Flash/Action Script by the author of this thesis, along with the publishing company Oxford University Press and has been adapted to XML and IMS-LD in the Open University of the Netherlands to be used as a base on educational gaming research and IMS-LD [149].

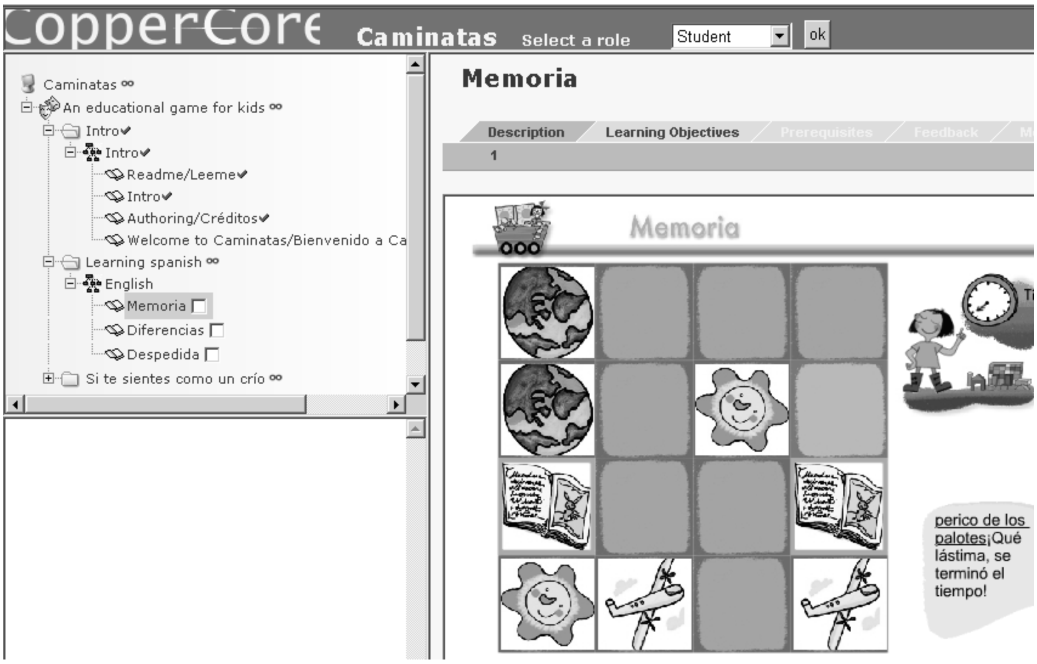


Figure 28. Educational game Caminatas in IMS-LD

Concerning IMS-LD version of Caminatas there are three possibilities of implementation: a) a mixed model Action Script – IMS-LD; b) a mixed model IMS-LD – Action Script; and c) a pure model IMS-LD.

The game consists of three very different moments or sections: Introduction, Game and Goodbye. Introduction involves setting-up the basic requirements of a run, shows the front page and requests the initial personal information of the player. In the Game section, a play of the game is run using the previous data. In the last section, Goodbye, these previously input data are used again, to personalize the closing message. The three sections are single files, stand-alone programmes, which send and receive the input data between them.

To solve this challenge of intercommunication between the modules there are three possibilities of implementation. In a) a mixed model Action Script - IMS-LD, the component SharedObject of Action Script is used as the built-in mechanism to store-retrieve information provided by Macromedia Flash in a proprietary extension called .SOL (Figure 29). In this case, IMS-LD is just a container of modules, a structural support for the internal execution of every module in Flash (Richards, 2005). The consequence is a one hundred per cent compatible modeling with specific insertions of code, regardless of their nature or original language. Java, Javascript, Action Script, Lingo an a large etcetera are valuable, therefore. This is the option taken in the provided example.

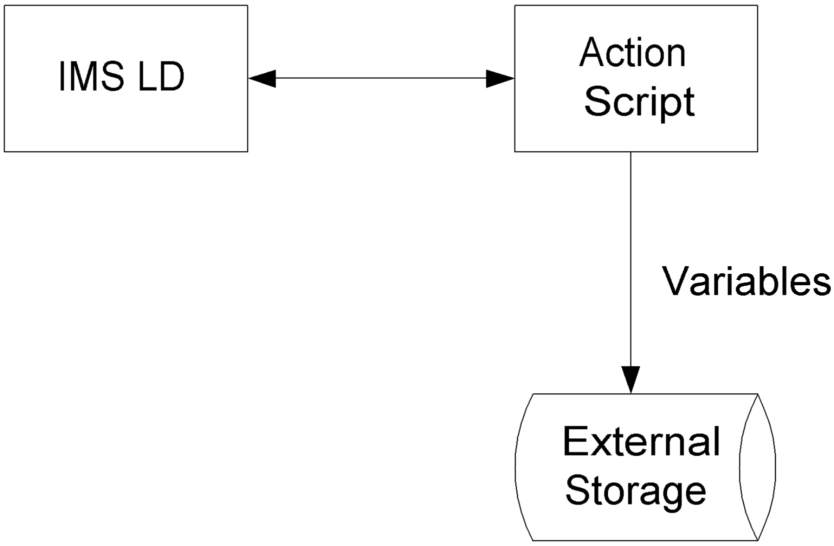


Figure 29. Model A

In b) a mixed model IMS-LD - Action Script, the information is sent from one system to another one using the variable interchanging through the heading of the modules and the methods GET y POST, usual ways in URL's communication and information linking with XML and CGI's (Figure 30). XML or any other Mark-up Language supporting these methods becomes the actual bridge between platforms. Afterwards, saving data could be done inside IMS-LD, using global variables and local variables and keeping the access in the Unit of Learning, or in Action Script, using the procedure explained in a).

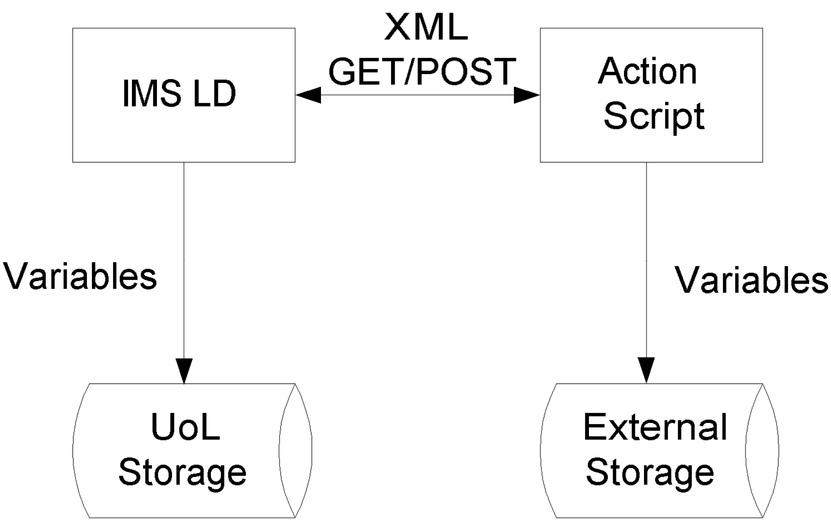


Figure 30. Model B

In c) a pure model IMS-LD, the game is full develop in IMS-LD, replicating in the specification the script already existing in Flash and using Action Script only to draw cartoons, if needed, and IMS-LD to store, retrieve and use variables and content (Figure 31). The IMS-LD core elements in Level B are main parts in this structure but the

GET/POST mechanism is passed by. Again, the information is still stored in the IMS-LD system but with no access as an external file.

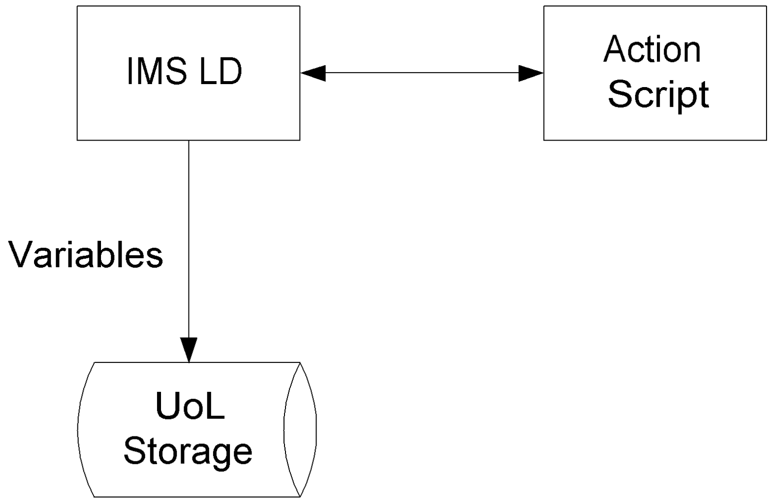


Figure 31. Model C

3.8 . Review of IMS-LD based on the integration of Units of Learning and Learning Objects internally modelled

This previous analysis leaves apart two main components. The first one is the interaction with an engine at run time, allowing the modification of the structure or the dynamic management of users, for instance. Establishing two main well different moments in the cycle of life of any electronic game as in any other computer application, design time and run time, modifying some structural or administrative aspects dynamically at run time is not possible so far in IMS-LD. This is an issue full related to the own essence of any IMS specification and something hard difficult to approach right now. Tools are evolving fast and maybe it is an expected and key improvement from engines/tools/applications chasing more flexibility in the creation and running of more powerful Units of Learning. Another alternative is to re-interpret the specification or extend it to support this kind the design on-the-fly.

The second one is the possibility of reading and writing external files. There is a difference between loading or retrieving a file and reading a file. Although an external file can be uploaded using the FILE property in IMS-LD, the use of data coming from that file in the development or the running is not so clear. For instance, in the setting-up of variables or providing dynamic contents for properties or fields. About storing information, IMS-LD doesn't allow a direct process to save data in an external file. Although the specification manages global properties to store contents outside an executed run, this information cannot be exported to any known and usual format.

This last issue on saving and loading information becomes an important one concerning the interoperability between some outcomes already created inside and outside an IMS-LD platform. The outsourcing of data generated in IMS-LD as well as the import of data generated outside IMS-LD into a learning structure ensures the proper integration of any Unit of Learning with the usual tools and documents needed in the daily costume of an user, such as managing students dossiers, valuating assessments, adding attachments, merging notes taken in different platforms and so on.

With the current version of IMS-LD, though, this last issue about storage can be solved using already existing languages commonly used to create resources, like web pages. For example, with HTML and JavaScript, the Fscomand function can be used to import, export and save information. With Action Script, of Macromedia Flash, the component SharedObject can save and retrieve information inside environmental variables, local or global ones. Also the function MMSave in the authoring system of Flash provides the possibility of storing an external ASCII-text file with the content of internal fields and variables in a CSV format.

Besides, if a higher level editor, language or programming environment is used to create resources, like Java or Lingo of Macromedia Director, for instance, there are some specific functions on storing/retrieving issues to do these jobs very easily.

Using these languages the storing problem is solved and information can be imported or exported inside the learning structure of an IMS-LD Unit of Learning. Although it is also true that the solution doesn't come from the specification. It would be much easier to work about this needed feature with some notation coming directly from the specification instead of passing by it to provide the solution.

Every component in an educational game can be modelled in both ways, mixed or pure. In the pure option, only with IMS-LD, several components of the Level A and the Level B of the specification can be used to model the structure of the game. Activities, roles, environments, method, acts, resources, for instance, and the communication among all of them, joining properties and additional conditions are available in order to get a right flow that represents the original script properly. But, also, in the mixed option, with IMS-LD and additional XML, a communication can be established between both using specific features of this language and the system of global properties, local properties and global elements. In this option, therefore, Level B is basic to manage all this elements. Following the list of main components of any educational game enumerated in this paper, 20 of 22 features can be modelled this way. Level A and Level B provide resources enough to sort it out.

The only two components, both technical, not completely affordable by now are 1) an engine at real time and 2) saving and reading external files. In terms of 1) an engine could be developed with any other language (Java, Action Script…) and to link the related information created with them to IMS-LD through XML code or using IMS-LD as a container of the main action and provider of learning and support activities. Because of the actual nature of the specification that clearly devides design time from run time, IMS-LD is not able to provide this feature by now. Maybe any of the current tools under development can support it in the upcoming releases.

In terms of 2) saving and reading external files, IMS-LD allows to upload an external file using the FILE property but it is not able to use the content of this file neither than save any property nor content in an external one. This feature can be developed complementing the IMS-LD action through an external language or program that saves and read the information in disk to send the result to the Unit of Learning managed by specification. Albeit IMS-LD cannot solve it directly it is possible to be done through a walk-around, taking advantage of the components and functions of others.

Regardless these two important issues, it is largely proved that IMS-LD can model 20 of the 22 analysed components. Just two of them, the engine, can not be directly affordable and just one of them has no obvious and quick solution. The other one, the external files, can use a combination of IMS-LD and other additional languages to find the closest solution to the challenge. This means a very clear outcome of the flexibility of IMS-LD to develop educational electronic games.

3.9 . Integration and Learning Objects: IMS-LD and eGames as re-purposed externally modelled resources

There is a growing interest amongst teachers in using eGames as a part of their lesson plans. A standardised, integrated approach to the sharing of such game-based lesson plans would allow teachers and educational technologists to compare and contrast Digital Game Based Learning scenarios, allowing best practices and lessons learned to emerge. Although games can be used as 'add-ons' in educational contexts, greater benefits can be attained by integrating games more fully into the educational process, i.e. by repurposing existing games to target the specific learning objectives. In this section we analyse this problem. We developed two possible solutions based on the integration and the interaction of games and learning scenarios. The first solution is based on 'pedagogical wrappers', where games are linked to eLearning flows of Units of Learning but without interaction and communication. The second solution sees a tighter integration which supports ongoing interaction and communication between the eGame and eLearning flow. We applied both solutions to a generic, already described game: Caminatas. We analysed the pros and cons of each solution and identify research topics for further research [150].

3.9.1. Introduction

This section concentrates on the use of generic games in education rather than on games designed specifically for educational purposes since: a) educational games are built for a specific educational goals and are focused on specific pedagogical aspects; b) educational games often tightly control the gaming flow, leaving the user in a passive role; c) Educational games are less widely available [151].

There are several pedagogical approaches that can be used inside and/or around a game, such as learning by doing, learning from mistakes, goal-oriented learning, role playing and constructivist learning [127]. However, using a game for learning purposes involves more than leaving students to play without a context or guide, hoping for results [130]. Exploiting the attraction of popular and widely available games in learning processes requires the appropriate scope, focus and context [152].

Examples of generic games used for learning include the Sims, SimCity, Flight Simulator, Pac-Man, FIFA, SuperMario Bros, Civilization, Rayman and Diablo II, for instance [134; 153; 154]. All of them belong to different categories of games. Following the taxonomy written by Crawford [155] focused on objectives and nature of the game, we find several well-defined categories, like skill-and-action, combat, maze, sports, paddle, race and strategy. Prensky introduces a categorization focused on pairs of opposite features, such as intrinsic versus extrinsic, reflective versus action, single-player versus multi-player.

The central question addressed in this section is how to re-purpose generic games for use in eLearning. We stress the importance of interoperability so that the lesson plans which result from such re-purposing can more easily be shared by teachers and educational technologists and used in different eLearning platforms and environments. We view lesson plans as combining pedagogy, and structured resources, files and links, to form Units of Learning (UoL) [52]. We define re-use as the action of taking an existing piece of content, learning object, learning activity, game or stand-alone module and using it out of its original context without modification. In contrast, re-purposing is defined as the re-use of some item for a different objective than the one it was created for. Finally, a learning object is described as any resource (text, link, video, audio, et cetera) used in an educational context and with learning objectives [52].

3.9.2. **Problem definition and two approaches**

3.9.2.a. ***Problem definition: re-purposing existing digital generic games and simulations within a pedagogical wrapper***

Currently, teachers sometimes use generic games in their lesson plans [132-135]. However, the use of games is often isolated from eLearning systems, leading to a disconnection between educational setting and game; a tighter integration offers pedagogical advantages. We illustrate these advantages with the following example. A teacher has a lesson plan explaining historical facts about the Fall of Rome and the collapse of the Western Roman Empire and he/she wants to tackle several activities: 1) Activity 1: Provide an overview of Rome at IV and V Centuries; 2) Activity 2: Make two groups in the classroom. One group will study the Romans and another one will study the Huns. Both will provide a detailed report and it will be uploaded online; 3) Activity 3: Each team studies the report of the other team; 4) Activity 4: Both teams will be divided in groups of five people and play the generic game ‘Rome. Total War. Barbarian Invasion’ [138] during a period of a week; 5) Activity 5: A final online chat will raise up some questions and stress some key aspects of the Empire, the Huns and this period of the History. The game,´Rome. Total War. Barbarian Invasion´, is a best-selling generic game distributed around the world.

The first approach will run the Activity 4 as an external game without any communication with the main flow. The students will extract an overview of the History as long as they play a game, but no connection is established with the Activity 3 or the Activity 5 inside the learning flow. The teacher, acting as a moderator, will focus all these activities and could stress some issues based only on his perception. In the second approach, in

Activity 3 a team could choose between several packs with three specific features to assign to the other group. For instance, the team of Romans could choose the pack number one consisting of amount of troops, strategy and time to assign to the team of Huns. The Huns could select a set of other three features to assign the Romans. The packs would have a mix of some good and bad features to keep a balanced starting point based on the previous knowledge of the participants acquired in Activity 2 and Activity 3. All these selections could be sent directly to the initial set-up of the game (Activity 4), as they could start playing with the pre-configuration. During the game, a list of values with the places visited, the conquests and the profit and loss of features is sent to the learning flow to provide a detailed report after the game. Based on this values the system could complement the weak points of each team providing extra access to some specific content.

We first examine how the game could be used in parallel with, but separate to an eLearning system, before turning to the advantages of a tighter integration. In addition, to illustrate the advantages of integration, we list several further examples along the two approaches.

3.9.2.b. ***Approach A: a pedagogical environment as a container of a game without any communication between them***

The game is incorporated to a learning flow but no further communication is established with it. It is executed as a stand-alone module embedded in a Unit of Learning (UoL). They can also be stored and run locally or remotely with a link from the UoL. Still, it is under the pedagogy expressed along the UoL as it is a part of the UoL. An example of a possible scenario could be the use of 'SimCity 4' [156] to work on Economics and the administration costs in a large city. A teacher could define an e-lesson plan with some activities focused on economical and administrative theory and a practical case using 'SimCity 4' as a simulator to recreate the scenario. In a certain stage of the learning flow, the student runs a local execution of 'SimCity 4' and use this experience to write an assessment on the topic afterwards.

Another possible scenario could be based on Asian Geography: a) groups of three students in a classroom have to enrol in a multi-player game of the Olympic Games hosted in a remote central server. Using the game 'Athens 2004' [140] every student chooses a different country of Asia; b) the game is set-up to count four sports. They play the first sport against the other countries and get a winner team; c) the students fill paper forms of his own country very in detailed and hang them out in the classroom and have time to go through them; d) they play a second sport in 'Athens 2004'; e) run of an electronic quiz on Asian Geography; f) they play a third sport, etcetera.

A third possible scenario could work on Dimensions and Perspective. The goal could be the understanding of a perspective: a) an introduction about 3D perspective in the daily life; b) theoretical description of how to build a perspective; c) play the game Tetris 3 [157] about fitting blocks in a 3D grid. This small game is inside the Unit of Learning as another packed resource; d) practical assessment on perspective.

Richards [158] describes this solution that leaves all the pedagogical workload to the pedagogical wrapper but not necessarily to the game itself. Along this way of integration, allocating or linking an external game inside a learning flow is straightforward because the game is used as another learning object, launched in a certain moment. Therefore, it works as another collateral activity without a real communication with the eLearning system, like reading a text, or posting to a forum or visiting a website. This is, something inside the learning flow as a learning object without return of any value to the flow, without any input from the flow and without any bearing in the flow, so. As there is no communication to and from the game with the wrapper none of them cannot influence in the set-up, the run or the progress of the other, or even of other activities. It is a kind of built-in or linked component without actual dynamic connection with its context.

3.9.2.c. ***Approach B: a pedagogical environment with a game as a fully integrated activity in the learning flow and some bi-directional communication between them***

The game is incorporated to a learning flow, it is able to receive information sent by the educational wrapper and it is also able to send itself information generated during the execution of the game to the wrapper. This way, both layers (main flow and game) can interact one to each other in runtime and modify features of both on the fly.

An example of a learning scenario in this approach could be based on General Knowledge and Adaptive Learning: a) run of a test with general knowledge questions for a single student using the generic game Trivial Pursuit [159]; b) there are three possible activities to be carried on afterwards. Depending on the result from the Trivial the student will be allowed to follow only one of them fitted to his/her level of knowledge.

A second possible scenario could be focused on Genetics and Contextual Feedback: a) a student works on the heredity of physical characteristics and he/she must define which consequences will come up after crossing three generations of people, focused on simple issues, as hair colour and eyes colour; b) the student fills an electronic form with his/her theory; c) the student runs the simulation game ´The Sims 2´ [160] and check if his/her theory is correct and if he/she gets what was predicted. If not, he can change this simulation on the fly and reconsider his/her theory, like in a lab; d) the assessment on the final result is provided in the main learning flow, together with a contextual feedback on the performance.

Along this way of integration the run of the game is not just a stand-alone one isolated inside a learning flow, it is a fully integrated part of the learning flow itself able to influence on it, to modify it and to adapt it along the running of the Unit of Learning. Therefore, there is a bi-directional sending of values between the game and the learning wrapper.

The suggested way of communication between these two elements is through an in-between layer created to allow the sending and reception of variables and values (Figure 32). This dispatcher would be the bridge between the pedagogical modeller and the externally programmed game facilitating the flow between them.

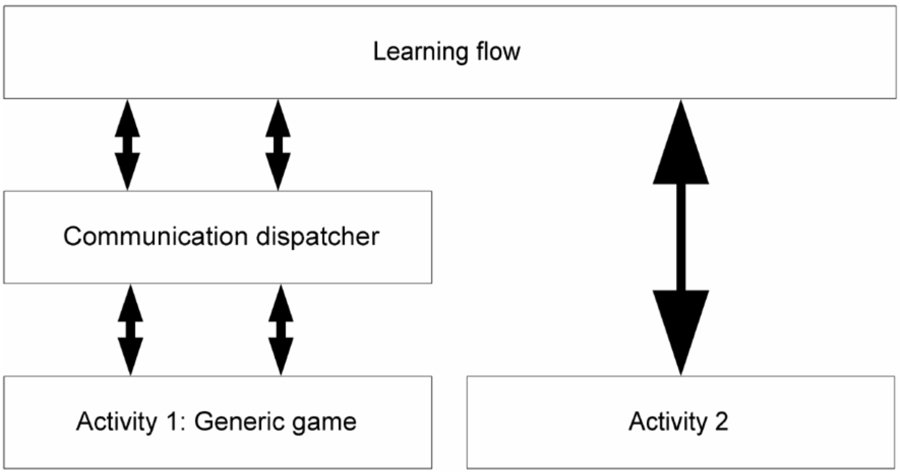


Figure 32. Architecture of communication between the learning flow and the game

Technically, we can compare this to the use of the methods GET and POST in a HTML page, that allow sending and receiving values between the HTML page and the outside. Java, JavaScript, Action Script, Lingo and a large number of programming languages can interchange values with the previously defined variables through this heading. Another comparison would be the suggested integration of SCORM packages inside IMS-LD structures while using a dispatcher [72].

3.9.3. **Case study: Caminatas**

3.9.3.a. ***3.1. Definition***

As aforementioned, Caminatas [142] consists of several stand-alone playable modules with basic data communication, grouped around a central main module to grant access and share tasks. It has personalisation features (user name), audio settings, and some adaptivity. As a reminder, the game consists of three very different moments or sections:

'Introduction', 'Learning language' and 'Goodbye'. The three sections are single files which send and receive the input data between them.

Within the 'Learning language' section two playable stand-alone modules are available run using the data gathered in the previous step. They can be played individually and in an isolated way. Also, it is not needed to play both to finish the game. The first module is based on the classical game of 'Memory' where the player must find pairs of identical images turning upside-down two cards at the same time. When a pair is found it is put aside. The game ends when all the pairs have been found or when the time is over (Figure 33). This game is not an educational game but can be used for educational goals, for instance, to improve memory, to identify signs, to train perception skills, to speed up simple arithmetical calculations, etcetera. The second module is based on a game of 'Differences', where the user needs to find a number of differences between two almost identical images. Again, this game is not an educational one, but can be used with this approach.

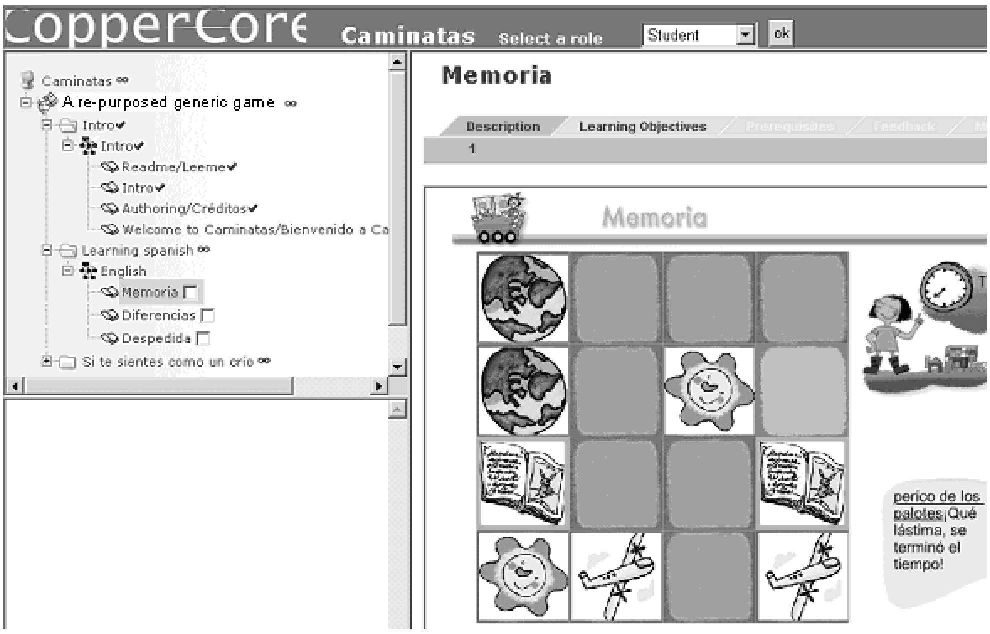


Figure 33. Generic game Caminatas re-purposed in IMS-LD to learn Spanish without any communication between the playable module and the educational wrapper

Neither of these two modules is an educational game per se, and they can be re-purposed for different uses and objectives. They can be considered as generic games and their goal and nature are provided by the context where they are used, also as a part of an eLearning lesson plan. Furthermore, every game could be integrated in a structure consisting of modules and modelled following a learning flow in IMS Learning Design with several elements, i.e, Plays, Acts, Activity Structures and Activities (Table 11). More specific, these two games were first programmed as puzzles for Primary School (with the

Oxford University Press publishing company) and re-purposed as educational games to acquire some basic Spanish knowledge becoming a part of a learning flow modelled in IMS-LD. We get a new use of existing learning objects taking care of the educational wrapper where they are built in.

Table 11. Modules of the integrated structure for Caminatas and related elements in IMS-LD

Integrated structure	Element in IMS-LD	Remark
A re-purposed generic game	Play	A Play consists of 1 or n Acts
Introduction	Act	An Act consists of 1 or n Activity Structures or Activities
Introduction:Introduction	Activity Structure	An Activity Structure consists of 1 or n Activities
Introduction:Readme	Learning Activity	An Activity consists of 1 or n Activity Descriptions
Introduction:Intro	Learning Activity	
Introduction:Authoring	Learning Activity	
Introduction:Welcome	Learning Activity	
Learning Spanish	Act	
Learning Spanish:English	Activity Structure	
Learning Spanish:Memoria	Learning Activity	
Learning Spanish:Diferencias	Learning Activity	
Learning Spanish:Despedida	Learning Activity	

3.9.3.b. ***3.2. Approaches to re-purposing***

The first solution (A) integrates the generic game inside a learning structure, modelled in IMS-LD in our specific case, but using it as a container of isolated stand-alone modules. This goes with no communication between the wrapper and the game. There is no value sent to the game from the pedagogical structure and no value sent back to the wrapper from the game. IMS Learning Design, or any other pedagogical modeller, becomes a mere skeleton where to insert already created games (Richards, 2004). This means that neither part influences the other as none of them knows about the other and they keep a simple link. There is no dynamic integration into the learning flow and the game remains as a static learning object inside a pedagogical structure and with no further possibility of information interchange.

As examples, this approach could be used in several scenarios: first, while learning the Spanish language, to play the modules as a support before and after two theoretical lessons. A second scenario could be focused on the ability to use the mouse pointer for elder people, providing some isolated exercises that help them to practise accurate clicking. In this case the two playable modules could be provided with no specific sequence and a simple text in the beginning and in the end of the UoL: a) short introduction; b) Memory and Differences; c) short closing. A third scenario could improve the perception skills working with a defined time to solve the same module several times along the flow but with a lower duration every time. For instance, a sequence of activities could be: a) introduction; b) Memory game with 60 seconds; c) Memory game with 40 seconds; d) Memory game with 20 seconds; e) theoretical background on perception.

If we use some of the existing IMS Learning Design editors, like CopperAuthor [4] or Reload IMS-LD Editor [94], for instance, to include a generic game or a simulation in a UoL only means a link to a resource or to a learning object. Simplicity of integration is the biggest advantage, as any generic game could be re-purposed without further modification through a simple link

The second solution (B) allows the sending and receiving of data between game and eLearning system. In this way, both can support each other and take into consideration some information from each provided for the most appropriate learning experience. Both, the game and the wrapper, can send and receive specific values stored in properties or variables to each other. Furthermore, the game becomes a fully integrated additional activity in the learning flow able to adapt certain values and features dynamically.

Following the three practical scenarios defined for the first solution (Spanish language, mouse pointer and perception) we could modify them to take advantage of the full communication. In the first one (learning Spanish lessons) we could use a previous test of knowledge to decide how much helping text in Spanish and which level of difficulty the game has (Figure 34). In the second scenario (using the game as a mean to improve skills on the mouse use for elder people) we could a) run a previous test to state the level of accuracy; b) depending on the result of the test, provide the game of Memory with more or less time to fulfil it; c) depending on the result of the game, provide a contextual feedback, stressing the importance of keeping active at elder age. In the final scenario (on perception) we could a) run the Differences game with medium-size changes between the two images; b) depending on the result of the game, provide a feedback about its accuracy and follow two possible itineraries, one repeating the game with large-size changes in the images and another one going through the Memory game with a similar protocol.

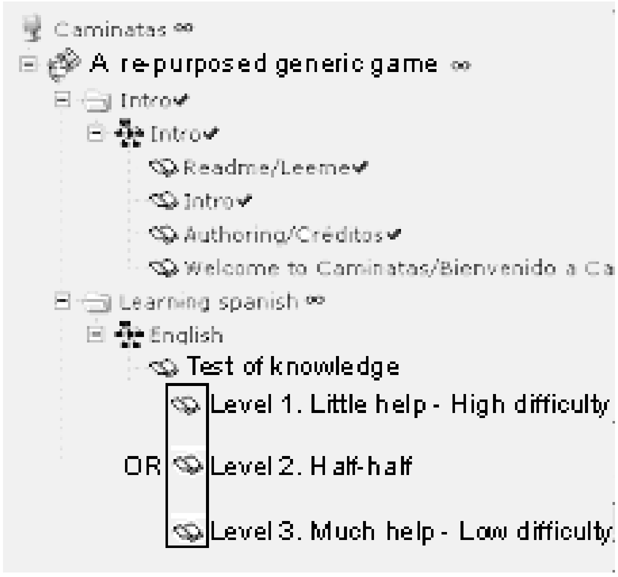


Figure 34. Re-purposing of Caminatas for learning Spanish with full communication between the playable module and the educational wrapper

The main advantage of this solution is the full control of the learning flow while using the game and the other way around, making the game another activity inside the lesson plan, and not an aside. The main disadvantage is that some modification in the existing game is required to allow the communication with the wrapper. This modification could require some programming skills from the learning designer or the teacher, or an editor capable of setting-up the variables and values to be used in the two pieces of software in an easy way. As not all the users have these abilities, and taking into consideration the current state of the educational wrappers, IMS Learning Design included, there is no easy way to implement this solution, by now. In addition, we would need a middle layer or dispatcher to establish the communication between the game and the educational wrapper. To summarize, the technical complexity and/or the lack of appropriate software are the major points against.

3.10 . Review of IMS-LD based on integration and re-purpose of Learning Objects externally modelled

The use of educational games within lesson plans is an increasingly common practice for teachers and educational technologists. Games connect specific content and skills with a friendly environment where the student is able to play, try, make mistakes and learn.

The possibility of re-purposing existing generic games and simulations in a didactic scenario providing a new pedagogical use to them becomes a challenge and a need in e-learning environments.

Digital games and simulations have a number of features that make them advisable to be used in learning processes. They improve certain skills and abilities, provide runtime feedback and allow the learning flow to be adapted on the fly. They also provide a good vehicle to establish social relationships and strengthen cognitive aspects in the player. Taking the generic game Caminatas as a base, we have re-purposed it from its original goal to an educational objective inside a pedagogical context modelled with the specification IMS Learning Design. The first solution is integrating a game in a pedagogical structure through a link, without interaction and communication. The main advantage in this approach is its simplicity for teachers and learning designers. The disadvantage is that the game is played as an isolated learning object with no consequences to the main eLearning system.

The second solution allows the interchange of values to and from the learning flow. The main advantage is that the game becomes another key part in the running lesson plan, opening up possibilities for adapting the learning flow on the basis of results and performance from the game, and able to use values from the eLearning system that could modify some features of the game. The main disadvantage is that this solution needs a technical approach which is not yet easily implemented.

Future research is focused on developing a software layer that can bridge the gap between an external module, game or simulation and an eLearning system based on IMS Learning Design. This will provide a more powerful environment for the integration of existing generic games and simulations into eLearning structures. This middle-layer should allow passing values between the IMS Learning Design structure and the external item and it should also allow some kind of interaction from one to each other. In addition, this dispatcher should be managed by a sort of editor or interface easy to use for teachers and learning designers without any need of a specific technical background.

3.11. Integration and Learning Objects: IMS-LD and fully integrated eGames

As we addressed before, electronic games and simulations (eGames) are a significant topic while talking about adaptation and learning. We take them as clear examples of Learning Objects. In addition to the motivational enhancements to the learning process provided by a playful and exciting user experience, eGames can provide adaptive learning content and fully personalised itineraries related to the user's behavior, previous knowledge or personal decision, to mention a few features on adaptive learning. Within eGames, the subgroup consisting of conversational games has several advantages worth examining: they are easy to understand and to model, they are well established and keep a long tradition in the game industry, and they provide an engaging and challenging experience where the content and the player's performance play a key role in the flow. In this line, the <e-Adventure> Project has developed a rich notation, engine and player to author conversational eGames (developed by the <e-ucm> research group, www.e-ucm.es). On the other side, IMS Learning Design (IMS-LD) is focused on providing a flexible specification to model several pedagogies, also focused on adaptation and, potentially, gaming. The main challenge comes when IMS-LD aims to integrate already existing games within the learning experience and establish a communication flow between the IMS-LD Unit of Learning and the external eGame.

In this section we show the relation between IMS Learning Design and <e-Adventure> when it comes to authoring adaptive Units of Learning containing eGames. We first describe the challenges of such objective and the several different solutions when integrating eGames in adaptive Units of Learning; we describe the content-centered authoring approach in the <e-Adventure> project, and its relation with the communication model between the IMS Learning Design specification and <e-Adventure> that makes possible a bi-directional influence on the adaptive learning experience. At the end, we describe a specific case study showing how we realized a fully operational adaptive IMS-LD Unit of Learning with an embedded eGame modelled with <e-Adventure> [161].

3.11.1. eGames and the integration with IMS-LD. Re-purposing eGames and levels of integration with IMS-LD

The re-purpose of already existing generic games or the on-purpose integration of specific educational eGames leads to a number of challenges when we try to integrate these external modules within an IMS-LD environment. Interoperability of the UoLs and of the eGames used is crucial to achieve the best educational goals in a personalised context, without compromising the rationale behind IMS-LD.

While the use of eGames is a fact within the teachers' community [132; 134; 135; 137], it is also a fact that this use is often isolated from eLearning systems and other information packages (e.g., IMS-LD, IMS-CP, SCORM). This issue stresses the disconnection between the educational setting and the eGame. A closer integration allows for pedagogical improvements and allows for a better contextualized learning path. As explained in previous sections, there are two main ways of integration between the eLearning wrapper and the eGame: a) the eGame as an embedded activity with no communication, and b) the eGame as a fully integrated resource, with a bi-directional communication with the environment and a state sharing. Option a) has been extensively described.

In the second approach b), in the previous Activity to the Activity-Game some actions (for instance, a quiz) can be carried out leading to a result. The score will be sent directly to the initial set-up of the game (Activity-Game), and it could start with an adaptive setting and behavior based on this input. During the game, a list of values of properties is sent to the learning flow to provide a detailed report after the game and-or influence the next action to take (for instance, choosing one learning path out of several possibilities). Based on these values the system could complement the weak points of each result providing extra access to some specific content.

Therefore, the game is a fully operational part of the learning flow, able to send and receive information to and from the educational wrapper, an IMS-LD UoL, via a communication dispatcher, and both layers can interact one to each other personalizing the learning experience, influencing the runtime and modifying features in both parts during the execution of the run. Therefore, there is a bi-directional connection between the eGame and the UoL, as previously shown in Figure 32.

However, it is important to design and implement this communication in such a way that it does not break interoperability. Otherwise, all the advantages of the formalization of learning methodologies provided by IMS-LD would be lost. In the next sections we describe the <e-Adventure> project, consisting of an authoring methodology and an educational game engine and we also study how this implementation can realize the integration of adaptive eGames in educational environments.

3.11.2. **The <e-Adventure> Project**

The main goal of the <e-Adventure> project is to apply a documental approach [162-164] to the development of educational graphical adventure videogames (often also referred as point and click adventure games). The idea is to allow an author without a strong technical background to produce and maintain an entire game as a document using an easy-to-understand language which is then fed to a compiler/interpreter (the <e-Adventure> engine) that produces a fully functional game.

The language is an XML dialect that the author uses to describe the environment, characters, objects and situations that form the game. The objective is to allow that author to build an executable game without needing a previous background in programming. All he/she would need is basic knowledge on how to use a computer, a text editor, a few notions of XML and a familiarity with <e-Adventure> syntax. In this section we will give a high level description of the philosophy behind the project. Further details on <e-Adventure> (formerly known as <e-Game>) are described in [165-167].

3.11.3. Authoring in <e-Adventure>

The authoring process in <e-Adventure> is summarized in Figure 35. Instructors involved in <e-Adventure> concentrate on preparing a document with the game's storyboard, and then to mark it up. To do so, they follow the writing conventions induced by the markup language. They also use the <e-Adventure> grammar to verify the correctness of the documents before feeding them to the system.

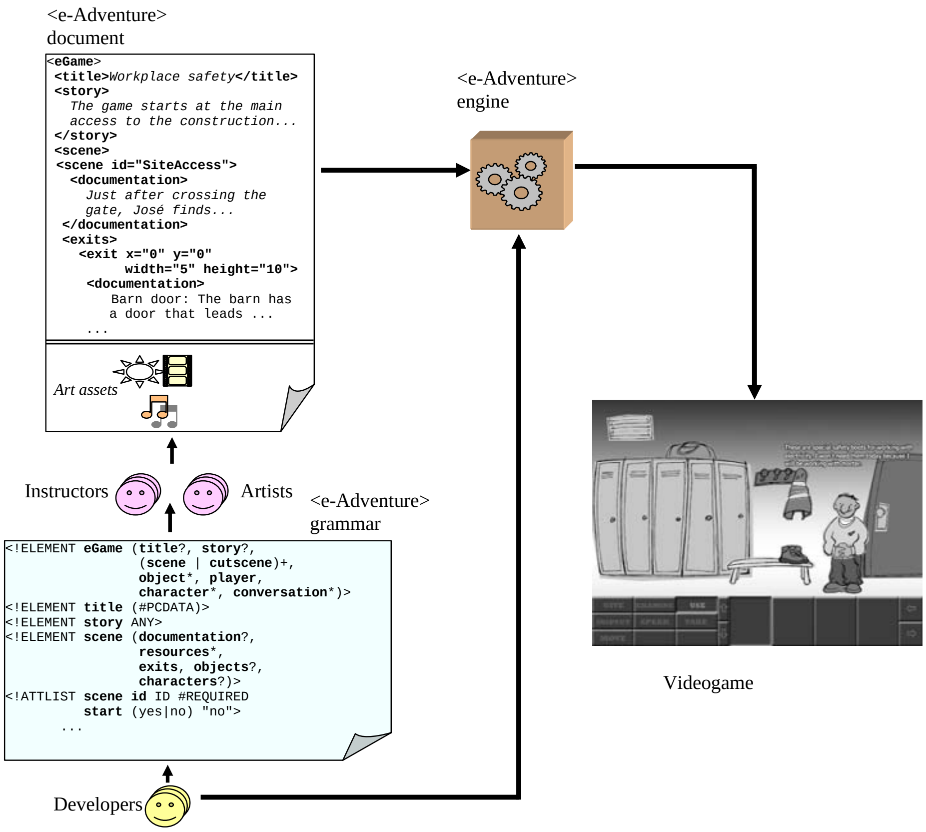


Figure 35. Authoring in <e-Adventure>

Writing a document in the <e-Adventure> language should not feel like programming, but more like writing a story. That is to say that the author does not specify how the characters move or how the lighting works, but what the actual content of the game is (scenarios, items, conversations, etc.). On the other hand, the documents should be human-readable and easily understandable by a computer at the same time. These needs suggest that XML technologies are a good candidate. XML is focused on describing hierarchically structured content, which is closer to our needs than a full programming language. Human readability is one of XML's design features [168] and it also provides mechanisms such as DTD and XML-Schema [169; 170] that allow the formalization of the language in a machine-readable way. This facilitates the author's work, by providing the means to verify the correctness of the documents before feeding them to the system.

The resulting <e-Adventure> documents should also refer to several art assets used in the production of the final videogame. For simple videogames instructors can provide themselves with the assets required (e.g. by using simple graphic editing tools). Nevertheless, in <e-Game> we also envision the participation of artists (graphic designers, musicians, etc) in the provision of the art assets for more professional videogames.

It is also relevant how <e-Adventure> explicitly involves developers (experts in computer science) in the authoring process. Developers produce and maintain both the <e-Adventure> grammar and the <e-Adventure> engine. Their involvement in the authoring process, that is explicitly promoted by the document-oriented approach in which <e-Adventure> is based on, makes it possible to react to the particular needs of each learning scenario. Indeed, developers can provide specific support for rendering the final games according with the directives of the instructors. They also can advice them in the use of the more difficult features of the <e-Adventure> language.

3.11.4. **The communication between <e-Adventure> and IMS-LD**

As described in Sections 3.7 and 3.9, the integration of any type of Learning Object, like eGames (implemented with <e-Adventure> or any other methodology), is a complex task and it raises a number of authoring issues. The root of the problem is that when a learner is interacting with a specific Unit of Learning, the specification demands that the player keeps record of the state of a number of variables, called properties that can be used to alter the path of the learning flow. On the other hand eGames are often analyzed in terms of game states, which may or may not be directly expressible in terms of IMS Learning Design properties.

It is necessary to provide the means to communicate and to translate the information used within the Unit of Learning and the information used within the eGames, which may require a strong programming background for the author.

However, <e-Adventure> supports a clear and narrow eGame model, in which information is stored and interpreted in a declarative fashion. The task of implementing the game is facilitated by the use of a domain-specific language that can be understood and applied without a programming background. The same ideas can be applied to the authoring of the information flow, thus allowing the non-technical author to specify the communications that should take place between the eGame and the Unit of Learning in a declarative fashion.

The rest of this section reflects this approach, and describes the documents that can be created to facilitate this communication.

3.11.5. Mapping UoL properties to <e-Adventure> game states

While authoring an adaptive eGame using <e-Adventure>, the game designer is required to implement that adaptation in terms of conditions over the state of the flags, since that is the mechanism used within <e-Adventure> to make conditional decisions at any point. Indeed, if the state of a number of flags is modified, the game can exhibit a completely different behaviour. If defined carefully, these different behaviours can correspond with different adapted versions of the same game. This behaviour is addressed by adaptation rules, like in Figure 36.

```
<adaptation-rule>
  <uol-state>
    <property id="PASSED_EXAM" value="true" />
    <property id="PREVIOUS_KNOWLEDGE" value="HIGH" />
  </uol-state>

  <game-state>
    <activate flag="FINISHED_CHAPTER_1" />
  </game-state>
</adaptation-rule>

<adaptation-rule>
  <uol-state>
    <property id="LEARNING_STYLE" value="EXPLORATORY" />
  </uol-state>

  <game-state>
```

```
    <activate flag="SKIP_TUTORIAL" />
  </game-state>
</adaptation-rule>
```

Figure 36. Example of adaptation rules. The first rule omits the first and most basic chapter of the game. Also, the initial tutorial describing the controls of the game and the tasks to be performed are skipped if a learner preference for exploratory learning has been identified.

This configuration file closes the gap that separates the internal representation of the state of a Unit of Learning and the internal representation of different initial game states, thus enabling the communication from the Unit of Learning towards the eGame. When the game is launched, the communication dispatcher depicted in Figure 32 checks the state of the properties within the Unit of Learning and sees if their value triggers a specific initial game state. In case it does, the dispatcher informs the game of which flags should be activated (i.e. which is the initial state for the game).

3.11.6. Mapping <e-Adventure> game states to UoL properties

Once the eGame has been designed and written using <e-Adventure>, instructors or learning designers can also prepare separate documents identifying those game states that are relevant from a pedagogical perspective and that should affect the state of the current Unit of Learning. When the <e-Adventure> engine is running, every change in the state of the game is notified to the dispatcher. The dispatcher, in turn, checks the new state against the list of pedagogically relevant states. If the state is listed, the dispatcher notifies the Unit of Learning to set the indicated properties to the corresponding values.

Each entry in this file is a mapping between a game state and a set of values for some of the properties present in the Unit of Learning. The game state is represented as a Boolean expression on the flags as used in the <e-Adventure> language itself. Meanwhile, the properties in the Unit of Learning that should be modified are expressed with a list of set-property elements identifying the property to be set and its new value.

Given the nature of this process, it is important to note that the separation between this mechanism and the definition of the game in terms of states conditioned by flags is wide enough to take an authoring approach in which the writer of the game and the instructor identifying the pedagogically relevant states need not be the same person. This mechanism can thus cater to a scenario in which the instructor is creating a game on his or her own and to a scenario in which the instructor is part of a team in which there are professional writers designing the game.

3.11.7. **Case study: The art and craft of chocolate**

In order to show the approach aforementioned we have developed an adaptive IMS-LD Unit of Learning with a eGame modelled with <e-Adventure> and a bi-directional communication flow resulting in an personalised learning path based on two inputs: the previous knowledge and the performance of the learner.

3.11.7.a. ***Basics and layout***

In this game, called "The art and craft of chocolate" (available at [60]), the final goal is to learn more about the world of chocolate from a practical side. The student has to know what is behind the ingredients and the history of this product to make tasty sauces that can match with both, a selection of meals and the expectations of demanding customers. The final goal is to get the highest satisfaction from the customers.

The eGame pursue several didactic objectives, focused on learning 1) how to make the right combination of the basic elements of chocolate to produce the base mix, 2) how to elaborate different chocolate-based sauces, and 3) how to marriage a few chocolate sauces with a selection of dishes. Every objective is related to one stage of the game. The description of every stage is in Figure 37. The first stage of the game (Library) deals with the origins of chocolate and the basic elaboration; the second stage (Kitchen) is more creative and allows for the elaboration of different chocolate sauces; and the third part (Restaurant) is a practical exercise with customers, where the learner should get the perfect marriage between the dishes selected by these customers and the available sauces. The final grade of the students depends on the satisfaction of every customer attended. In fact, all the previous steps are only a preparation for the live test with clients. To provide a more playful environment, the player decides on his own when he is ready to face the customers, although his level of previous knowledge influences the access point. There is no minimum number of sauces required, although there is a restriction: (s)he needs to collect and sort several ingredients and objects, and to make several sauces before being granted to go to the last part (Restaurant) for the live test.

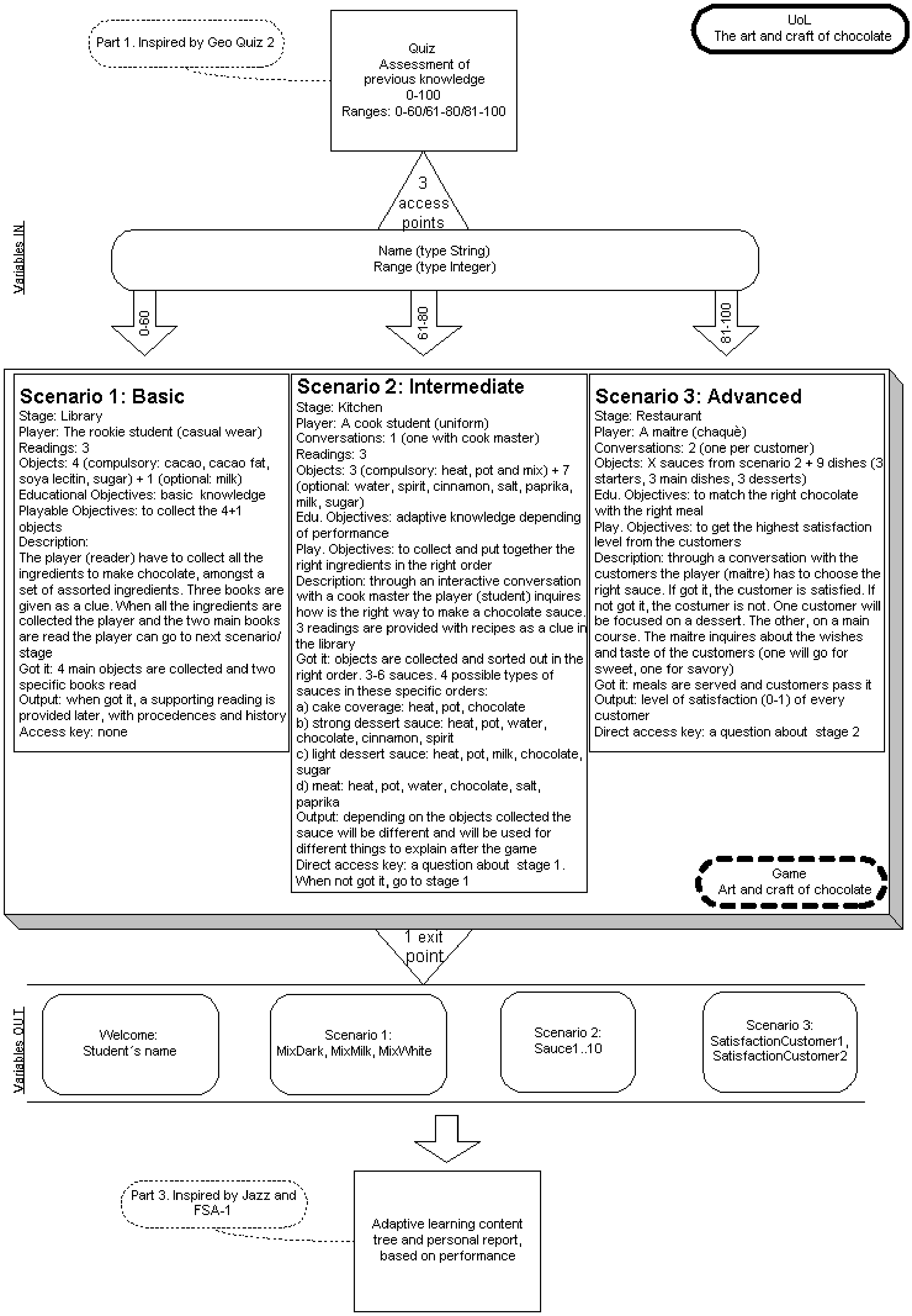


Figure 37. Structure and dependencies of the UoL. Definition of the 3 stages in the game

The structure of the eGame and the UoL where the eGame is embedded is as follows: the game itself is embedded in a full Unit of Learning with a previous quiz and post-adaptive learning path (see Figure 37). Depending on the score out of the quiz, the student accesses directly to one of the three stages, where there are control questions. When the answer to one of these questions is not right (s)he receives the right answer and is sent back to the previous stage. Also, once the student is in one stage he can go back to the previous ones as long as he fulfils the access requirements. When the control question for a direct access to stage 2 is right, the student automatically receives 2 mixes (dark and milk). The student is granted to come back to the stage 1 to make more mixes at any time. This communication flow is shown in Figure 38. When the control question for a direct access to the stage 3 is right, the student receives automatically 3 sauces. The student is granted to come back to the stage 2 to make more suaces at any time.

Once the game is over several variables are sent back, stating the satisfaction level of both customers and which of the possible sauces were actually prepared. The UoL takes these results about the performance of the learner and provide an adaptive learning path out of three possible alternatives.

3.11.7.b. ***Authoring the IMS-LD Unit of Learning***

The game is embedded in a full Unit of Learning consisting of three parts (Figure 38). The first part is focused on a quiz on the topic where the learner has to answer five multiple-choice questions to establish his-her previous knowledge. Based on this score, the learner will be granted to access to a different area of the second part, the game. During the game, the performance of the learner is stored in several variables that will be sent back to the IMS-LD part of the UoL. Once the game is finished, and based on the values of these variables, the UoL will provide one or another learning path to follow. In this case, the adaptation is present in the three parts of the UoL, based on previous knowledge and actual performance, although it could be author to other different types of inputs. The first part, the quiz, is based on the UoLs Geo-Quiz 1 [171] and they deal with properties, conditions and calculations in order to define the questions and get the appropriate results out of the user's answers. It also works with adaptive content, showing and hiding different areas depending on the actual moment of the run, and providing an adaptive feedback based on the performance of the user.

The third part, the adaptive learning path after the game, is based on and Geo-Quiz 3 [171] and it defines up to three different possible itineraries as learning activities. All of them are hidden until the game is finished. Afterwards, only one of the activities is shown, directly related to the performance of the user during the game. In this case, the adaptive learning flow is linked to different contents focused on several aspects to learn about the topic, and it tries to provide the best information for every performance of every user.

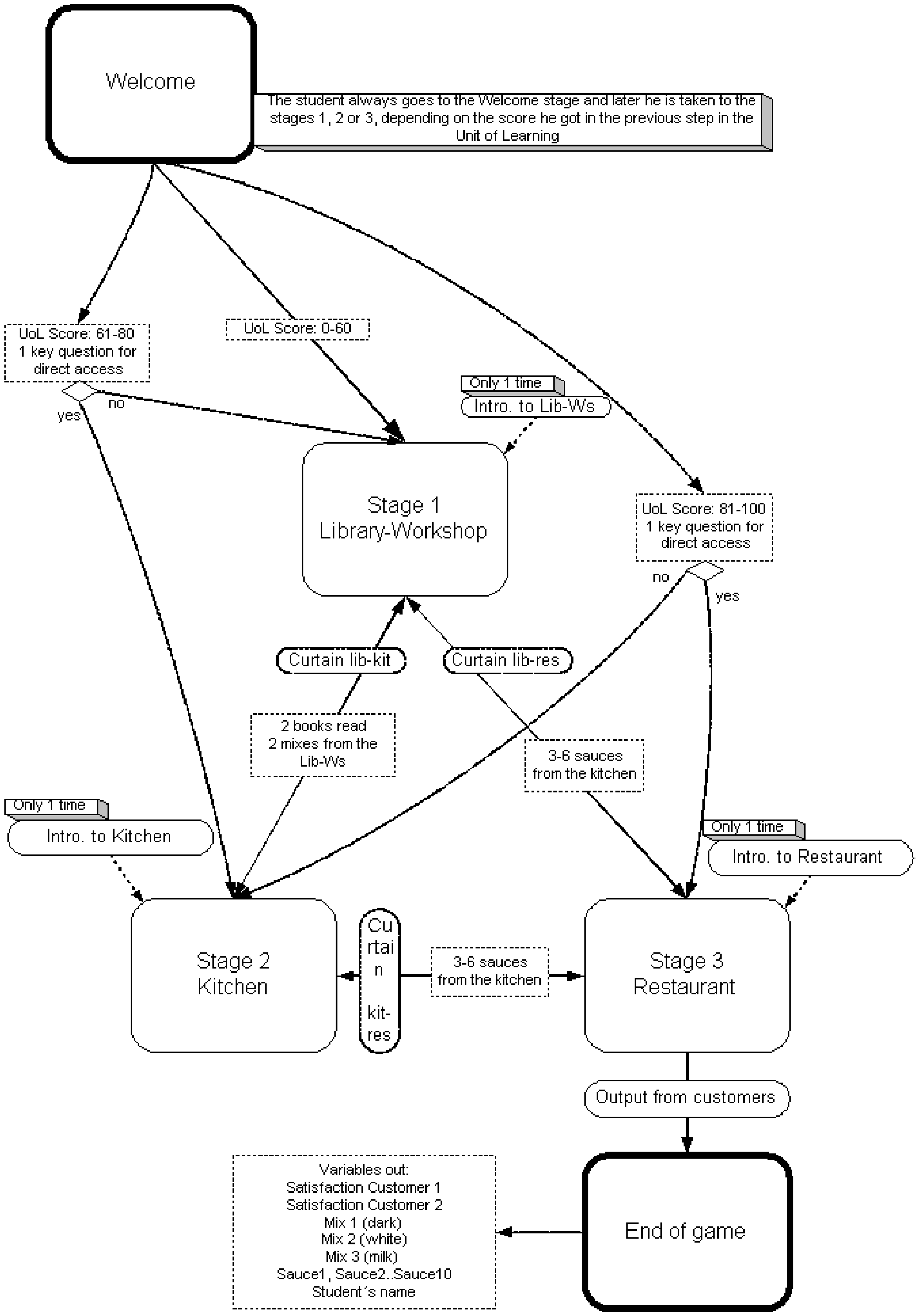


Figure 38. EGame layout and communication flow.

3.11.7.c. ***Authoring the adaptive <e-Adventure>***

The <e-Adventure> game itself can be defined in parallel, because the adapter can handle the necessary transformations to align (i.e. to translate) the values and variable names used in the Unit of Learning and in the <e-Adventure> game.

The process of writing the adventure starts with the elaboration of the storyboard following a number of guidelines that will facilitate the markup process following the principles of descriptive markup [172]. After the storyboard is approved, there follows an iterative process in which the storyboard is marked up and reviewed and in which the different art assets are produced in parallel. The development of the game itself is a well-studied development process and it is described with greater detail in [166].

However, it has already been mentioned that the communication with the IMS Learning Design raises two additional authoring issues: it is necessary to adapt the values used inside the Unit of Learning and the <e-Adventure> documents for both directions of the communication.

On the one hand, for the communication from the Unit of Learning towards <e-Adventure> with adaptation purposes a document using the syntax described before was created. In particular, the first part of the Unit of Learning sets two very specific properties indicating the level of the student according to his or her responses to the quiz questions. The input configuration file for the adapter indicates which internal flags in the eGame should be activated to alter the behaviour of the game so that the simpler parts can be skipped.

On the other hand, for the communication from the <e-Adventure> engine to the Unit of Learning with assessment purposes, a different document was created. The eGame keeps Boolean flags indicating which dishes with what sauces were served to each customer. The output configuration contains rules that associate dishes delivered with punctuations that will be reported to the IMS Learning Design environment, thus affecting the path followed in the third part of the Unit of Learning.

3.12 . Review of IMS-LD based on the fully integration of external Learning Objects

Since the beginning of IMS-LD in 2003, its relationship with the rest of surrounding technologies such as Learning Management Systems, other non-IMS specifications, databases or simple stand-alone executable modules made in any programming language has become a challenge. A deeper stress has gone to improve the pedagogical expressiveness but not to achieve such this kind of technical practicalities. The connection between IMS-LD and <e-Adventure> and the bi-directional flow of properties that are able to modify the learning flow and the information in both sides shown in this section is a contribution in this direction.

However, the introduction of this communication mechanism raises new and important issues when it comes to authoring the adaptive courses and/or Units of Learning. With the objective of allowing teachers and Learning Designers to model rich lectures, full of interactive learning objects, educational objectives and tasks it is necessary to provide the proper authoring mechanisms.

As it has been described, there are several tools and approaches to IMS Learning Design authoring that cover the authoring needs as far as the description of the general learning model is regarded. Concerning authoring of the adaptive eGames, <e-Adventure> facilitates the definition of conversational eGames for every person with a very low technical threshold. Following a simple and well-documented process, any person can design and implement small to medium-sized conversational games with special features that enhance their educational value, including adaptation and dynamic assessment. It is relevant to note that the development process can also be scaled to bigger organizations creating longer and more complex games, although in the context of this work the target audience are instructors creating smaller games.

The two authoring processes are related and their joint use is relatively simple, although still technically challenging for the average user. There is a need of defining some properties and variables to exchange information and affect the flow of the Unit of Learning or the eGame. There are no graphical editors for either the <e-Adventure> games or the additional configuration files that enable this communication and thus the technical gap could be significant.

Given this situation, two main lines of work are foreseen. The first one points to the real use of such this authoring setting for adaptive educational games with actual teachers and students. This goal is double-sided: teachers should practice the process and the making of IMS-LD Units of Learning with embedded eGames made with <e-Adventure>, and students should follow the created course. Both user-groups should provide extremely valuable feedback based on practice.

Second, the development of a graphical editor that makes the creative process easier and suitable for every creator-teacher-learning designer, no matter his technical expertise. This is especially relevant for the definition of the rules that relate the state of the learning process with the internal state of the eGames, which is technically challenging and error-prone and represents the main obstacle in the implementation of the approach.

Section 3 depicts the integration of Units of Learning in several contexts. It takes the basis provided by Section 2 on general issues of IMS-LD and focuses on this specific challenge. The main conclusion underlines the need for a communication mechanism between IMS-LD and the others, as well as for the inclusion of specific elements and structures within the IMS-LD information model which would facilitate such both-ways interchange of data

4. **Adaptive learning**

This section dives into one of our two main research lines: the adaptation of Units of Learning to specific needs of the users involved (namely, learners and teachers, but also tutors, admin staff and others). We represent different types of adaptation with IMS-LD and find potential improvements and weaknesses of the spec. In addition, in the Annex (Section 10), we model five different use cases which show actual learning scenarios that provide a very useful feedback and which support a set of partial conclusions to be added to the ones already depicted by the two other core Sections of analysis, 2 and 3.

4.1 . **General research framework: definition of adaptation**

Recent research has addressed the definitions of adaptivity and adaptability, both focused on personalised learning [2; 173-175]. In summary, adaptivity is the ability to modify eLearning lessons using different parameters and a set of pre-defined rules. In contrast, adaptability is the possibility for learners to personalize an eLearning lesson by themselves. These two approaches go from machine-centered (adaptivity) to user-centered (adaptability). However, we contend that there are a vast number of stages in between which define a grey area, with some adaptivity and some adaptability [176]. In practice, it is quite difficult to isolate one from the other due to their close relationship [177]. Furthermore, all the in-between stages are also personalised learning as they enable the dynamic adaptation of several features in a course, a lesson plan or a Unit of Learning. As a result, we view the concepts not as two opposite corners from which to look at personalised learning, but as describing a wide range of approaches taking the best of each. Hereafter, we use the word adaptation to cover the various approaches.

From the user interface through the eLearning resources to the learning process there are many aspects to take into consideration for effective adaptation. From the early eighties, where Computer Based Training was used to fully control the flow of a learning process [178; 179], to the concept of Adaptive Guidance, which provides rich information and a diagnosis to help the learner to take effective decisions about his own learning [180], there is a wide collection of approaches to adaptation in eLearning. For instance, to incorporate the tutor as a key factor in the adaptation process [181], or to build a blended system strongly supported by AI agents [182]. All od them are based on the proposal of personalised learning adaptation to the context of each student in order to stimulate his learning process and to encourage his involvement in this process [183-185]. These approaches also hold that the best learning performance comes from personalised instruction [186]. This does not necessarily imply that a user/student should keep full control over his training, because this would mean that 1) the student knows what is the best for him along a learning script; 2) the student is aware, knows and controls all the contributions that he can make to his own process; and 3) the student is able to carry out the right decision when all this information is collected [187].

We define adaptation in eLearning as a method to create a learning experience for the student, but also for the tutor, based on the configuration of a set of elements in a specific period aiming to increase of the performance of pre-defined criteria [181]. These criteria could be i.e. educational, economic, time-based or user satisfaction-based. Elements to modify/adapt could be based on content, time, order, assessment, interface and so on.

4.2 . **Adaptation by whom?**

The Adaptive Hypermedia Systems approach indicates that the best adaptation is the one that a user cannot see; the one where the cognitive load of the user is reduced to the minimum or even zero [177; 188]. For example, a user is able to follow an introductory section in a course only the first time he logs-in, since that section is hidden from the second login onwards. This means that the adaptation must be done by the system automatically, depending on the user's performance. From this point of view, mainly based on adaptive hypermedia Web systems, there are two well-defined inputs: user and set of rules This approach is focused on micro-adaptation: monitoring the learning behaviour of the student while running specific tasks and adapting the instructional design afterwards, based on quantitative information [189; 190]. On the one hand, there is a provider of some information based on behaviour (the student). This means a non-voluntary action where the information is collected from the provider without his knowledge or agreement. On the other hand, there is a decision maker (the set of rules) who takes the collected information and decides further movements based on it.

Although this is an interesting approach, there are some drawbacks on this view of the involved inputs. First, the input of the student as a decision maker, who progresses from one side, being a non-voluntary provider of information, to the opposite side, stressing the cognitive load in the final decisions about the learning process. We agree that sometimes a student cannot decide what is the best or the most convenient for him at a certain moment, because of lack of training or of a broader vision on his studies [187]. But we also hold that, leaving apart his interests and opinions, this approach dismisses his motivations, wishes and personal drives (i.e. feelings and emotions). All of them are important parts in the learning process. Second, the input of the teacher as one of the participants involved while building adaptive courses is something to take into consideration. His inputs, feedbacks and contributions do not end when the course is delivered. A teacher can and should modify the learning flow of a student or a group of students as long as the course takes place. He should make a good balance between the time and the effort invested and the actual outcome, though.

Usually, adaptation is focused on the student. Adaptation involving tutors is clearly also possible, although it could be more time and resource consuming because the teacher should provide a personal or group guidance instead of giving collective lectures or in addition to them. A third element in the list of adaptation-makers is the set of rules that provides adaptation taking different inputs from the different stakeholders. Taking into account these three factors (learner, teacher and set of rules) instead of only one (learner) a full and richer adaptation system can be setup. The final goal for all of them is to provide the best fit to personalised learning objectives. Furthermore, we count up to four inputs in a balanced formula for adaptation: a) the learner, when some information is taken from his behavior and performance and when the cognitive load is reduced to the minimum or even at all [177-179; 188]; b) the learner, when he contributes with his/her own personal decision and he takes over the next step to take along the learning path [180]; c) the teacher, when he also contributes with his/her personal decision and he evaluates the personal situation of the user related to the full learning experience [181]; and d) the set of pre-defined rules made by the learning designer/author (usually, the same teacher) [182]; this kind of adaptation engine is usually expressed in the way of AI agents and nested conditions. The inputs a) and b) provide some personal needs and drives to learn; the inputs c) and d) take care of the didactical quality and of the learning efficiency.

Finally, when we refer to the learner, this could also imply a learner-group, where a set of objectives, activities and processes is setup for a whole.

4.3 . Adaptation and feedback in eGames

In order to define a link to the other main topic of this dissertation (integration), we stress the importance of adaptation on two subtopics: eGames and feedback. The relevance of eGames as one of the most flexible means to establish a fully rich relation with Learning Objects has been described in the previous chapter. The connection with feedback is described in this section, as feedback is one of the most used and effective mechanisms for adaptation [191].

Feedback proved to be critical for adaptive learning [192] as it provides support on the educational process and motivation [193; 194] and feedback is also an important feature of games [127]. Garries et al. [193] define a cycle in games that involves several loops of repeated judgment, behaviour and feedback and they point out that feedback is a critical component to regulate the learner’s motivation. Also Csikszentmihalyi [195] stresses the importance of feedback in his classical definition on flow theory. Kernan &

Lord [196] state that specific feedback based on goals commitment increase the effort, the performance and the motivation of the learner. People are expecting some reaction on their actions and efforts and they become disappointed when they do not receiving it, resulting in a decrease of their motivation and performance [197].

Prensky [127] supports the use of feedback to support adaptive learning and game-based learning as a way to provide useful and immediate information to the learner about his performance. Baer et al. [198] underline that immediate and contextual feedback improves learning and reduces uncertainty. Kirriemuir [199] highlights that instant feedback invites and allows for exploration, experimentation and it stimulates the curiosity. Kiili [200] and Chen et al. [201] write that games should provide instant and appropriate feedback in order to improve the performance and the level of adaptation.

However, not all authors agree on the positive effects of feedback. In their literature review on computer and video games for learning, Mitchell and Savill-Smith [202] state that the one and same feedback mechanism could be perceived in different ways. They refer to a study by Halttunen and Sormunen [203] who used an educational game to support learning of information retrieval. After an evaluation, the effectiveness feedback users received from the system was generally seen to promote learning significantly. Feedback concerning the performance of one's own query and the chance to reformulate the query and to further evaluate the effect of changes on performance was seen as a highly motivating and learning advancement. However, there were also students reporting that their attention was fixed on performance and they tried to improve on their results mechanically, without analysis and reflection of their preceding queries and results. Here, the feedback tempted searchers to pay attention to the performance measures achieved, rather than on analysis of the search task situation and strategy.

Educational games, and simulations even more, are usually played in an unpredictable way by the player/s. This unpredictable behaviour can provide some values based on performance on which a final analysis back to the learner will be given, although this feedback could be too complex or not specific enough to make it useful or even easy to understand or to apply [204]. In addition, once some feedback is provided its use by the learner is uncertain, meaning also a reduction of the performance [205].

In computer applications, there are numerous means to provide feedback to a user (teacher, user or system): visual, auditory, or force feedback just to name a few, still others will emerge. In the definition of Mason & Bruning [206] feedback is defined as any message generated in response to a learner’s action. It implies that there is an interactive flow between the learner and the system, coming from some information collected or generated by the first one [207]. Furthermore, this information flow is seen as a series of frequent inputs and not as a single one, because it is a part of the entire learning flow [208], and all these inputs become a crucial part of the adaptive learning itinerary.

In context of Human Computer Interaction (HCI), feedback can manifest itself in different ways, but in the bulk of studied task environments it mostly refers to some sort of information presented to the user after something is done. This can be an indication that a certain action actually has been carried out (or not) or providing qualitative feedback about performance, amongst other things. Since the widespread introduction of Graphical User Interfaces, it was common to stick to guidelines such as for example the ones that Apple started to provide from 1992 and on [209]. One of the guidelines was, and still is, to keep users informed about what's happening by providing appropriate feedback, and enabling communication with the application. When a user initiates an action, there should always be an indication that the application has received the user's input and it is operating on it, since users want to know that a command is being carried out. A few years after GUI's and WYSIWIG interfaces became common however, Gentner and Nielsen [210] wrote an influential article not meant to discard the Macintosh interface guidelines, but to explore alternative approaches to computer interfaces. One of the issues here was to consider bringing some flexibility into the feedback and dialogue provided by the system. They pondered that rather than always providing the user with feedback on activities, the computer should be more flexible in the amount of feedback it provides. Initially, the computer could provide detailed feedback to familiarize the user with operations and instill confidence. Later, the feedback could be scaled back over time and restricted to unusual circumstances or times when the user requests more feedback. Jansen et al. [211] also support the use and benefits of cautious visual feedback in learning processes to get a better performance more focused on the task and to look after four main purposes: monitoring and adaptation, social awareness, group processing and engagement, mediation of development.

Through appropriate feedback, the learner is able to receive some information concerning the way he acts and learns. This enables him to assess his own progress regarding his goals an actions [127; 193], and he is able to make a consequent choice about the next action to take or even about the strategy to follow [197], stressing the influence of feedback on the personalised learning process.

4.3.1. **Types of feedback**

There are several ways to provide feedback, and it can be based on several adaptive features: the learner's performance, the learning history or the learning goals [127]. Furthermore, it is fully driven by the learner's activity; his data is collected as input, after analysis by the system information is returned back to the learner as output [207; 212]. Mory [192] describes two main types of feedback: instructive and informative. In short, instructive feedback is related to knowledge domain and informative feedback is related to the context where learning happens. While the instructive one leans on a corrective intervention on the learning process, the informative is focused on self-regulation. In addition, there are four main types of indicators in informative feedback: 1) related to performance, 2) related to process, 3) related to social interactions and 4) related to environmental interactions. Performance feedback is the most common one in eGames, but not the only one.

4.3.2. **Destination feedback and eGames**

There are still other types of feedback. An example is what the Apple Human Interface Guidelines [209] refer to as destination feedback. It refers to the phenomenon that when the user drags an item from its place to a destination, the application provides feedback that indicates whether it will accept that item. Destination feedback should not occur simply because your application is drag-aware; rather, it should depend on the destination's ability to accept the type of data contained in the dragged item. This type of feedback informs the user of the possible actions that can be taken. In a certain way the user is taken by the hand, because the choices are being limited. Everyday examples where this principle is being applied are software installation wizards, extended help-options and greying-out menu-items that don't permit using them at a given moment. The latter (very common in Apple, Microsoft and Macromedia/Adobe products) provides a context-sensitive feedback in the interface, by only permitting relevant/possible actions. Feedback by greying-out of items is one example of what is referred to as externalizing information. The externalization of information makes information readily available on the interface, and it is not necessary to remember that information. In other words, only recognition, not recall, is needed for task performance, and this relieves working memory [213; 214]. In the opposite situation, when no such features are provided, a user has to internalize the information himself, and store this information in his/her memory. Zhang used various problem solving tasks as material showed that externalizing information can be useful for cognitive tasks; the more information is externalized, the easier it is to solve the problem [213].

It is easy to see how the concept of destination feedback can be interesting in game-based learning. Providing guidance or assistance in complex situations and trying to relieve the working memory of students so that they can devote attention to development of proper strategies is interesting in this context. But one can also question the assumed positive effects that this kind of feedback can have. Perhaps learning with a feedback-based interface is more volatile and difficult to transfer to other situations. This is undesirable when learning or gaining insight itself is the aim. We pondered that having this kind of destination feedback (which is extremely common across a wide range of applications) might users to behave less proactive and lazy, and do less thinking before you act. And exactly this can be undesirable when learning or gaining insight itself is exactly the aim of the learning task.

Research by O'Hara and Payne [215] provides support for this notion, stating that a too strong reliance on external information leads to negative effects regarding planning and transfer of skills. They drew a distinction between plan-based and display-based problem solving, which can be seen as analogue to what happens during internalization vs. externalization. During plan-based problem solving one has to construct problem strategies and subsequently use detailed problem strategies from long-term memory. Display-based problem solving on the other hand makes little use of learned knowledge but relies on interface information. Plan-based activity leads to a shorter solution route, because steps are planned, and no unnecessary steps are taken, while a display-based strategy involves more steps because of more searching and less planning. Also, Svendsen [216], who used the Towers of Hanoi problem showed that a high-cost interface yielded improved understanding of problems.

The notion that too much feedback could be counterproductive while playing a game based on planning, led us to do a case study with real learners. In the coming section we present the differences between having, and not having destination feedback and the drawbacks of making use of it, while playing an educational simulation focused on planning.

4.4 . Taxonomy of adaptation types based on content and elements

A combination of the following proposals on adaptation could support the performance of every role in an eLearning process [183; 217].

Traditionally, three types of adaptation have been proposed:

1. Interface-based (also called adaptive navigation and related to usability and adaptability) where elements and options of the interface, are positioned on the screen and their properties are defined (color, size, shadow, etc) [173]; this is closely related to general customization and supporting people with special needs which influence personalisation, such as color blindness or poor hearing, for instance [218].

2. Learning flow-based, where the learning process is dynamically adapted to sequence the contents of the course in different ways. The learning path is dynamic and personalised for every user, but even also for every time that the course is started (also called run or instance), so that the user can take a different itinerary depending on his performance.

3. Content-based, where resources and activities dynamically change their actual content, as in Adaptive and Intelligent Web-Based Educational Systems based on adaptive presentation [219; 220]. For instance, the information inside a learning activity can be classified in three levels of depth, and every level is shown based on a number of factors.

The first block of adaptation types are the base for the following ones. Additional kinds of adaptation are [56]:

4. Interactive problem solving support, which guides the user on the next step to take in order to get the right solution to a problem. The guidance could come from an online or offline tutor or from a predefined set of rules.

5. Adaptive information filtering, taking care of appropriate information retrieval that provides only relevant and categorized outputs to the user [221]. Although this type provides adaptive information, it could be considered as an external facility linked to a learning activity and not as a real part of that learning activity itself.

6. Adaptive user-grouping, that allows ad hoc creation of groups of users and collaborative support on carrying out specific tasks. For instance, as a result of a pool of questions, two groups with beginners and advanced users are made.

We extend the classification further with:

7. Adaptive evaluation, where the evaluation model, the actual content and the running of a test can change depending on the performance of the student and the guidance of the tutor [181].

8. Changes on-the-fly, the possibility to modify/adapt a course on-the-fly by a tutor or the author of the course in run-time [222], moving beyond the previous types which are set-up and defined in design-time [223; 224].

In a literature study, we identify eight different kinds of adaptation being carried out in eLearning systems [225]. All of them use various inputs provided during the learning process and aim to tune the activities and actions of the learner to get the best learning experience as possible [226]. A wide and consistent set of rules of dependencies among users, methods and learning objects is needed to describe these eight types of adaptation, and moreover their possible combinations [46].

4.5. IMS Learning Design and adaptation

IMS-LD [81] provides a modeling language able to design executable Units of Learning (UoLs) [68; 85; 227]. There are two main approaches to create adaptive UoLs: first, an initial analysis by Towle and Halm (2005) sees the adaptation fully modelled inside a Unit of Learning, and describes four areas in IMS-LD where some kind of adaptation could take place: environment, method, roles and activities. Second, Van Rosmalen and Boticario (2005) examine the external adaptation of a UoL, making modifications to both the internal elements of the UoL and the playing layer through which the UoL is delivered (player).

We now examine how IMS-LD can be used to represent each of the eight types of adaptation aforementioned.

4.5.1. Interface based

Interface adaptation is based on menu options, navigation facilities and visualization facilities. This issue relates to the user interface provided with IMS-LD players such as the player included with CopperCore [228], the Reload Player [229] and Sled [230]. The current generation of these tools do not provide facilities to allow interface adaptation in run-time, although Sled can be customized during the set-up using stylesheets. Current IMS-LD IMS players cannot change the size and position of their panels or working areas, the definition of their windows or any other navigation facility. These players cannot change basic features, like font-size, font-color, font-type or alignement, either. There is a distinction between the external wrapper of the Unit of Learning (player) and the actual Unit of Learning itself with real content and learning activities. The player is the tool that allows for interpreting and viewing the Unit of Learning. Although interface adaptation cannot be carried out with the current players, some kind is possible inside the Unit of Learning, if we use two resources: DIV layers and environments. We can work with DIV layers that can be shown and hidden in run-time by any of the main participants in the learning process (i.e., user, teacher, set of rules). Inside a DIV layer we can define different options and/or look and feels of the same content, meaning a de facto interface based adaptation. In the same line, we can use several environments to provide different setups (i.e., contents, approaches, views) related to the same Unit of Learning, leading to a final personalised interface. Although neither of these two solutions (DIV layers and environments) is based on the external wrapper/player they can provide another view to interface adaptation.

4.5.2. Learning flow based

The description of an adaptive learning flow is mainly based on four out of the five different available elements of IMS-LD at Level B [68; 231]: properties, calculations, global elements and conditions. In addition, monitoring services can be added to track users' behaviour and allow the teacher to adapt the flow dynamically. An example of these features is provided by Learning to Listen to Jazz (all the examples can be found at [171]). A student can learn something about four different Jazz styles in a sequential way, and he can choose between a thematic itinerary and a historical itinerary, following different milestones in the course. An additional example is GeoQuiz 3 where the activities are defined by the performance of a student after answering an evaluation form. Depending on the final score and the related level acquired, one or another activity is shown. A final example is Cándidas II showing full learner control by the student, who directly selects which is the best method to study a lesson among four different options.

4.5.3. **Content based**

The content of an activity needs a resource linked to the element Activity Description. Although this link cannot be changed at run-time, three other elements can be modified dynamically:

- the content inside an XHTML resource, defining classes and DIV layers that can be hidden and shown based on certain parameters;
- the content of pre-defined properties/variables, that can be replaced with other content typed-in on the fly;
- the content of an activity can be adapted switching showing or hiding one of several linked environments.

Two examples of the use of environments are Learning Activities with Conditions, where a student decides the granularity level that he wants and From Lesson Plan to IMS-LD Level B, where again a student takes control and switches on and off the audio support of the UoL. Finally, the aforementioned Learning to Listen to Jazz provides contents linked to several Activity Descriptions and related environments, progress-based.

An additional way of content-based adaptation is the modification of contents linked to fixed resources and based on external tools. For instance, a resource linked to a wiki service hosted outside an IMS-LD UoL could adapt its content dynamically, based on users', tutors' or authors' contributions.

4.5.4. **Interactive problem solving support**

This kind of adaptation could be considered as an extension of learning flow based, with the appropriate definition of properties and conditions modeling the itinerary, and the incorporation of a monitoring service allowing the tracking of the learning process of the student, making ad hoc remarks and changing the process as needed. These changes can be carried out 1) by modifying specific arguments by the tutor, 2) by the execution of specific design-time rules, or 3) by a combination of both mechanisms. An example is What is Greatness where the tutor moderates the contributions of a group of students on an open question, providing access to the next step when the tutor thinks that the current one is finished. A further example is Free Style Assessment where a tutor and a student carry out a commented open evaluation of an assessment. The tutor is entitled to close and block every step and to provide contextual feedback.

4.5.5. **Adaptive information filtering**

IMS-LD is not designed to provide adaptive information retrieval. Some rudimentary facilities are available through the index-search service. More practically, IMS-LD could point out to an external searching service providing the container for the run of this application and also for the visualization of the results.

4.5.6. **Adaptive user grouping**

User management has two approaches, one based on role creation, where the users are assigned to, and one based on the creation of the users itself. Using the management system provided by several tools and engines – Coppercore, Reload, CopperAuthor [232] – once the UoL is published, the administrator (maybe the teacher himself) can add and delete users and assign them to a specific run of that UoL. This means a de facto group [233]. However, the dynamic creation of roles after the publishing process is not currently possible. Once a definition of roles or stakeholders is available, and a run of a UoL is defined, specific users can be added to, or removed from, any of these groups and these users can play the run. Some representational facilities are available in IMS-LD to support creation of groups (min-persons and max-persons) and although assignment of users to groups can be achieved, fully automatic on-the-fly creation of groups may require additional representational devices.

4.5.7. **Adaptive evaluation**

Taking the performance of a student in a Unit of Learning as input, a full set of parameters can be stored in local properties to be used in the adaptation of formative or summative evaluations. As we have already explained related to Geo Quiz 3, certain actions and answers of a user can be allocated into variables pre-defined in design-time and they can also be interpreted in run-time following a set of rules. In this way, both the evaluation system and the content itself, and even the interpretation of the results, can change for each user. An example is Quo Builder 2 where a questionnaire can be fully set-up with questions, answers, thresholds and feedback being defined in run-time. Again, the main obstacle to overcome is the run-time modification of the skeleton itself, such as the ordering, grouping and numbering of questions and answers; something not possible so far with the current state of tooling. However we can define a wide set of questions that can also be hidden and shown on demand, providing a top-down 'simulation' of adaptive extensibility.

4.5.8. **Changes on-the-fly**

Every UoL has three clearly different steps in its own life-cycle: design-time, publishing-time and run-time [227]. With the current tools, once a UoL is published it is not possible to change structure, method or definition of basic parameters (such as conditions or properties, for instance). Of course, if a UoL is so designed, a tutor is able to change the way a student perceives the course and the flow: 1) tutors can update the content, based on pre-defined content or on new contributions; and 2) tutor can also influence the learning itinerary, uploading files, showing and hiding content elements and structure elements, etc. This means that a tutor is able to change things on the run, as long as he had previously defined that possibility in design-time. This solution comes with a high expense on implementation and support, though. An example is the already mentioned Quo Builder 2 where a tutor makes the set-up and initialization of an evaluation form within run-time, which is subsequently filled by students.

4.5.9. **Review of IMS-LD based on types of adaptation**

Adaptation is a quite complex process taking into account several stakeholders and inputs: user, teacher and set of rules. Also the right balance between the cognitive load of users and teachers and the non-voluntary actions that can be taken as inputs in the set of rules defined inside an engine should be composed. IMS Learning Design seems a promising expressive language that allows for several types of adaptation. We can use the elements available at Level B - such as conditions, properties, calculations, global elements and monitoring services- to model and run rich and adaptive Units of Learning.

The possibilities for adaptation supported by IMS-LD are diverse. From the eight types of adaptation described we identify three levels of support: a) learning flow, content, evaluation and interactive problem solving support are well supported, although they could be improved with specific structures focused on adaptation (i.e., modification of the learning design on the fly); b) user grouping, interface adaptation, adaptive evaluation and full modification of a course on-the-fly are partially supported; c) last, as some pending issues with no support at all are dynamic modification of learning structure and method in run-time, and adaptive information filtering and retrieval. Some of this lack of support leans on the current state of tooling, and not on the specification itself, though.

Nevertheless, with several types of adaptation, like content and information retrieval, also it is possible to provide specific support on adaptation, i.e. linking a learning activity to an external tool that provides a related service and keeping IMS-LD as a container for external adaptation. To this extent, adaptation comes from outside IMS-LD although the learning design acts as an integrator. In conclusion, with the appropriate support, IMS-LD can build adaptation and rather flexible learning experiences for every stakeholder.

4.6. Taxonomy of adaptation based on inputs and roles. Main types of adaptation and related representation with IMS-LD

The eight types of adaptation described before could be divided into three groups [2]. The first group a) points to learning flow, content and interface [56; 173; 218-221]; the second group b) is based on interactive problem solving support, adaptive information filtering and adaptive user grouping [56; 221]; the third group c) consists of adaptive evaluation and changes on-the-fly [222; 223]. The groups b) and c) could be considered as subgroups of a), as they make use of the types in a) to define and feed themselves. Hereafter, we will focus our research on the first group of adaptation (learning flow, content and interface based), explaining what these approaches consist of and how to represent them in IMS-LD.

All of them take several inputs and roles, as shown in Figure 39: on the one hand, learner, teacher and learning designer; on the other hand, behaviour/performance, learning style, personal decision/cognitive load, method and set of rules. All of them synchronized around the learning system/engine. This structure of roles and inputs also takes into consideration the relationship between several users acting along the learning process, and it underlines the relevance of collaboration in learning (i.e., group of learners, group of teachers).

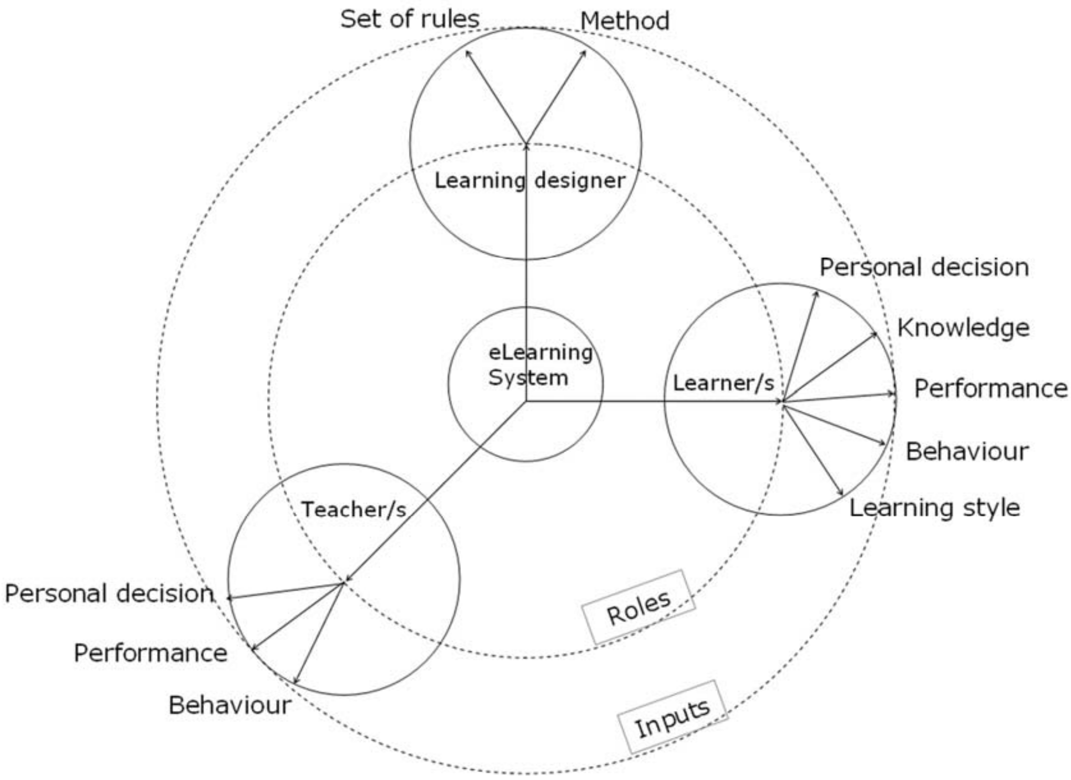


Figure 39. Taxonomy and relation of inputs and roles in adaptive eLearning

This conceptual model could be translated into an IMS-LD based model (Figure 40), which takes elements and components of the specification to express adaptation in three differents views. Option A makes use of one only activity to work with environments, layers and items related to every single role: all in one, with a set of conditions and rules that orchestrate the play. Adaptation takes place within the activity. Option B defines one activity for every role, all of them bundled in one only activity structure. It can also use environments to enrich the scenario. In this case, every rol has adapted content and resources in his/her activity. Therefore, adaptation comes before any activity actually runs. Last option, Option C, is defined in a single activity where every role plays a role-part, along with environments, layers and items. In this case, adaptation comes in design time and it is structured through the actions that a role takes in a certain moment. This option can be merged with Option B to define role-parts as activities of the same activity structure.

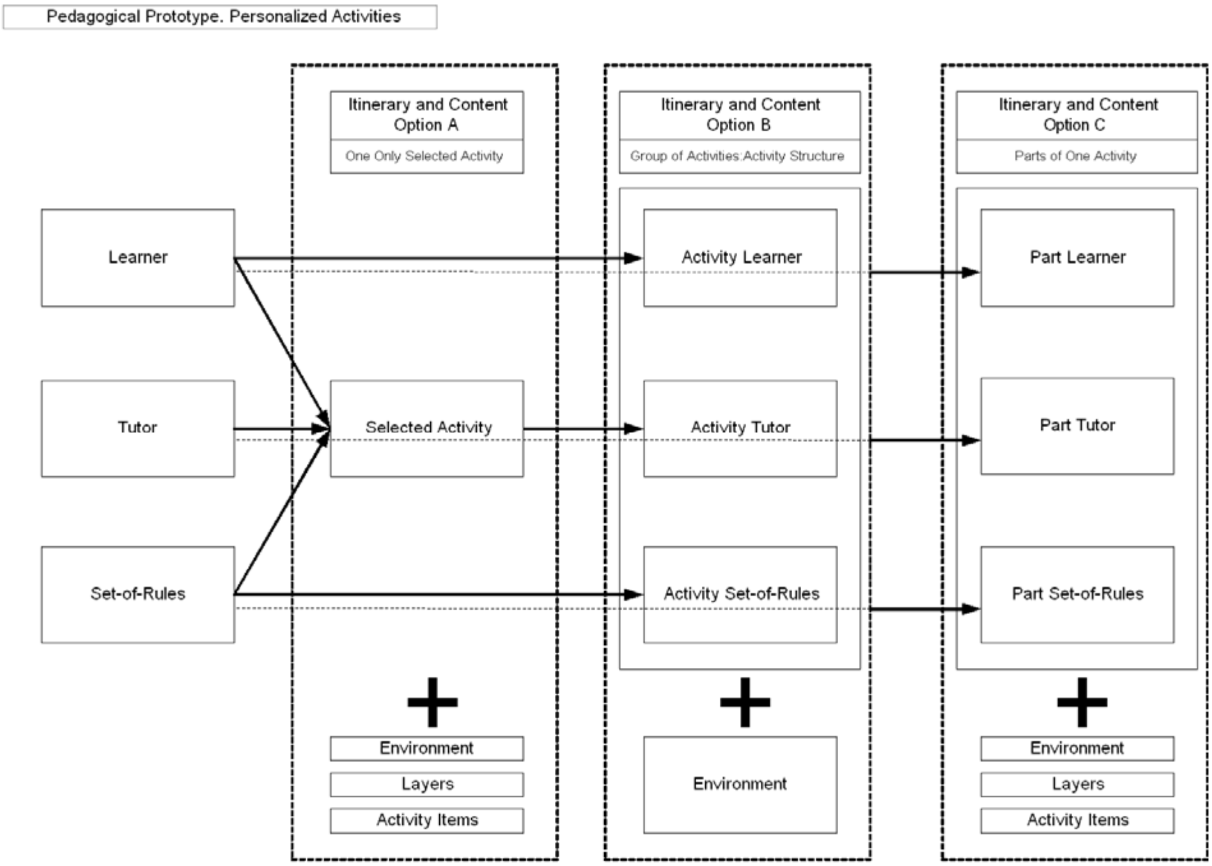


Figure 40. Conceptual adaptive model translated into an IMS-LD based model

4.6.1. **Adaptation based on learning flow**

The modification of the learning flow as long as the Unit of Learning is being executed is one of the most frequently used types of adaptation. Taking the flow as a base, the Unit of Learning provides different activities, resources and services, depending on these four inputs during execution (i.e., the learner´s behavior and performance, the learner´s decision, teacher and set of rules). The activity structure in an IMS-LD UoL is defined using plays, acts, activity structures, learning activities, support activities and environments. We can also use the property of visibility to hide and show these elements and to adapt the learning flow. In these cases the property works as a flag, switching on and off the referred elements. We now show five scenarios and their related implementations of learning-flow based adaptation focused on the several possible inputs. The pseudo XML code shown onwards is an abstraction of the IMS-LD original source, concentrating on the key elements of the specification needed for a more self-understandable explanation; all the examples can be found at [171].

4.6.1.a. ***The set of rules modifies the learning flow taking the user's behaviour as an input***

Scenario: one activity is shown the very first time that a user logs into the Unit of Learning, and it remains hidden from the second time onwards. User and the set of rules are the inputs involved. In this context, and onwards, the concept *user* is focused on the role of learner. However, it well could concentrate on the role of teacher, as long as both can provide their inputs of behaviour, performance and the others afore mentioned.

Implementation: we create a personal local property (Prop-Firsttime) (type Boolean) and it is initialized to 0. When the user logs-in the first time the property is set to 1. A condition shows a specific activity (LA-FirstActivity) only when this flag property is 0:

```
<locpers-property identifier="Prop-Firsttime ">
  <title>PropFirsttime</title>
  <datatype datatype="boolean"/>
  <initial-value>0</initial-value>
</locpers-property>
```

```
<if>
  <is>
    <property-ref ref="Prop-Firsttime"/>
    <property-value>0</property-value>
  </is>
</if>
<then>
  <show>
    <activity-structure-ref ref=LA-FirstActivity"/>
  </show>
```

```
    <change-property-value>
      <property-ref ref="Prop-Firsttime "/>
      <property-value>1</property-value>
    </change-property-value>
  </then>
  <else>
    <hide>
      <activity-structure-ref ref=LA-FirstActivity"/>
    </hide>
  </else>
```

4.6.1.b. ***The set of rules adapts the learning flow based on the user's performance***

Scenario: the behavior of a user is a possible input. Also, the performance and the cognitive load of the user during an activity could adapt the learning flow. This example, GeoQuiz3, provides a general quiz on Geography with five questions and multiple answers. The user gets a score, an average and an accuracy measure. The subsequent activity to be studied by the student depends on these results, and it is taken from four possible activities, including the repetition of the task if a certain threshold is not reached. Therefore, user and engine are the inputs involved.

Implementation: this example guides the learning flow of the student based on his performance and on a set of pre-defined rules. Also, the pre-defined algorithm interlaces properties, activities and conditions to get the final result. First, a set of properties is arranged:

```
  <locpers-property identifier="Value1">
      <datatype datatype="integer"/>
      <initial-value>0</initial-value>
  </locpers-property>
  <locpers-property identifier="Question1">
      <datatype datatype="string"/>
      <initial-value>Select</initial-value>
      <restriction type="enumeration">Select</restriction>
      <restriction type="enumeration">Malasia</restriction>
      <restriction type="enumeration">The Moon</restriction>
      <restriction type="enumeration">Canada</restriction>
  </locpers-property>
```

Second, every question is included in an external XML file, using the global element set-property. In this case, HTML code and IMS-LD code are combined:

```
<html>
    <td>Where is the Mare Tranquilitatis?</td>
    <td><p>
        <set-property ref="Question1" property-of="self" view="value"/>
    </p></td>
</html>
```

And third, the conditions are established in the learning design to check the results and to define the adaptive feedback and the next activity to be undertaken. For instance, by changing the completion value of a question, as follows:

```
<if>
    <is>
          <property-ref ref="Question1"/>
          <property-value>The Moon</property-value>
    </is>
</if>
<then>
    <change-property-value>
          <property-ref ref="Value1"/>
          <property-value>1</property-value>
    </change-property-value>
</then>
```

4.6.1.c. ***The user himself modifies the learning flow based on his personal decision***

Scenario: in Learning to Listen to Jazz the user can choose the learning itinerary out of two possible paths: Historic and thematic. The user can swap between both at three different points in the learning flow. The activities already done in one path remain in the same state when the user moves to the alternative path. Therefore, they are the same activities but with two different ways of study. In this case, the adaptation comes from the user, based on a pre-design of the course by the author/tutor.

Implementation: both paths are predefined and are shown and hidden depending on the value of the Boolean property SelectionOfRoute that the user can change on request:

```
<if>
    <no-value>
          <property-ref ref="SelectionOfRoute"/>
    </no-value>
</if>
<then>
```

```
      <hide>
            <activity-structure-ref ref="AS-Thematic"/>
            <activity-structure-ref ref="AS-Historic"/>
      </hide>
  </then>
  <else>
  <if>
    <is>
      <property-ref ref="SelectionOfRoute"/>
      <property-value>Thematic</property-value>
    </is>
  </if>
  <then>
      <show>
            <activity-structure-ref ref="AS-Thematic"/>
      </show>
      <hide>
            <activity-structure-ref ref="AS-Historic"/>
      </hide>
  </then>
```

4.6.1.d. ***The teacher modifies the learning flow of the user***

Scenario: the teacher monitors the performance of a user or of a group of users and he decides which activities should be shown and hidden and in which order. The adaptation comes from the teacher, taking into consideration several other inputs from the user or group of users.

Implementation: two things are needed. The first one is a monitoring service S-Performance defined in an environment in the learnign design that allows for the observation of every user. The second one is the definition of a set of flag properties (personal or role properties). These properties show and hide activities, structures and environments to the end-user/s, e.g. FlagForActivity1. The definition is made in the imsmanifest.xml file; the actual view and set-up is made in an external XML file using global elements (view-property and set-property):

```
  <environment identifier="E-Performance">
    <title>You can watch the performance of every user</title>
    <service identifier="S-Performance">
      <monitor>
        <role-ref ref="Tutor"/>
        <title>Tracking personal performance</title>
```

```
      <item identifierref="R-Performance"/>
    </monitor>
  </service>
</environment>
```

```
<set-property ref="FlagForActivity1" property-of="supported"/>

<view-property ref="StudentPerformance" property-of="supported"/>
```

*4.6.1.e. **Integrated approach***

A last scenario is based on the integrated decision of several of the previous scenarios agreed by consensus. More than one input (user, teacher, set of rules) are taken at the same time and the final decision leans on one of them. For example, a teacher can take the suggestion of a learner, his behavior, his performance, the recommendation of a set of rules and make the final decision. Alternatively, the engine takes the role of the teacher when making the final decision. Or even the student takes the suggestions of the teacher and of the engine and decides what to carry out next.

4.6.2. **Adaptation based on content**

In the previous section, we saw that a learning flow is mainly focused on the sequence of the activities in a Unit of Learning. However, content based adaptation is focused on the information of every activity, and on the activity itself. There are two main approaches for content based adaptation: flag properties and content of properties.

*4.6.2.a. **Flag properties***

This approach is focused on the use of flag properties that switch on and off a certain information layer (such as a DIV layer in XHTML).

Scenario: a student follows a questionnaire. The right answer to every sequenced question is the key to read the next question (example GeoQuiz1). Depending on the answer, some support information is shown.

Implementation: the actual definition of the layers (Answer1_Wrong and Answer1_Right) and the information inside them is defined in external XML file/s. These files are linked to the imsmanifest and identified as resources of type `imsldcontent`; they also have the layers to be shown and hidden and they use global elements:

```
<html>
    <h1>Question 1/5</h1>
    <p>Where is the Eiffel Tower?
```

```
      <blockquote><b>A</b> Paris</blockquote>
      <blockquote><b>B</b> Brussels</blockquote>
      </p>
      <p>Your answer is:
          <set-property ref="Answer1" property-of="self"/></p>
          <div class="Answer1_Wrong">
            <p>Choose another answer.
            '<view-property ref="Answer1" />' is not right</p>
          </div>
          <div class="Answer1_Right">
            <p>Congratulations! It's the right answer</p>
            <img src="eiffel.jpg"></img>
      </div>
  </html>
```

The definition of the method is in the imsmanifest.xml file. The definition and initialization of the flag properties and the learning activities are also done in this file, as well as the management of the visibility of the DIV layers described in the external files. When QuestionTrue1 turns to 1 the first activity question1 is finished:

```
  <locpers-property identifier="QuestionTrue1">
      <datatype datatype="boolean"/>
      <initial-value>0</initial-value>
  </locpers-property>
```

```
  <learning-activity identifier="question1">
      <title>Question 1</title>
      <activity-description>
            <item identifierref="res-question1"/>
      </activity-description>
      <complete-activity>
            <when-property-value-is-set>
                  <property-ref ref="QuestionTrue1"/>
                  <property-value>1</property-value>
            </when-property-value-is-set>
      </complete-activity>
  </learning-activity>
```

```
  <if>
      <is>
            <property-ref ref="Answer1"/>
            <property-value>A</property-value>
```

```
    </is>
</if>
<then>
    <hide>
          <class class="Answer1_Wrong" />
    </hide>
    <show>
          <class class="Answer1_Right"/>
    </show>
    <change-property-value>
          <property-ref ref="QuestionTrue1"/>
          <property-value>1</property-value>
    </change-property-value>
</then>
```

4.6.2.b. ***Content of properties***

The second approach to adaptation allows for the modification of the content of a property that has been pre-defined inside an activity. In this case, the property of visibility remains always on, but the content of the field changes. We need two steps, therefore. One is at design-time, making the definition and configuration of the property, done in the learning design of the imsmanifest file. The second step, at run-time, involves changing the content of the property, made in an external XML file. We use the global elements set-property and view-property to configure and see the content of the field/property. In the example GeoQuiz3 used above, the property with the adaptive feedback (prop-feedback) is shown after the completion of the form:

```
<td><p>Your adaptive feedback is: <view-property ref="prop-feedback"
property-of="self" view="value"/></p></td>
```

Depending on the final score, the content of the property (prop-feedback) is different, though:

```
<if>
    <and>
          <greater-than>
                <property-ref ref="score"/>
                <property-value>49</property-value>
          </greater-than>
          <less-than>
                <property-ref ref="score"/>
                <property-value>76</property-value>
          </less-than>
```

```
    </and>
</if>
<then>
    <change-property-value>
          <property-ref ref="prop-feedback"/>
          <property-value>Well done! You are in Level 2</property-value>
    </change-property-value>
</then>
```

Another possible scenario is set-up when a teacher changes the content of some fields dynamically while executing the Unit of Learning. In Quo Builder2 the student can see the questions and possible answers in a questionnaire as long as the teacher defines them. The teacher can also design the basic configuration of the form: the general welcome messages, the adaptive feedback and the scoring system. At the end, it becomes an interactive and dynamic evaluation test, modified at run-time.

4.6.3. Adaptation based on interface

Interface based adaptation is quite different to content based adaptation. This is the reason why we analyze this type of adaptation in third place, although we list it in an earlier position in the taxonomy depicted in Sections 4.4 and 4.5, Content adaptation is based on the information inside an activity that is shown and handled. Interface adaptation is based on options, navigation and visualization facilities. In [2] the authors state that interface adaptation is not possible with today's tools for IMS-LD, such as CopperCore Player [119], Reload IMS-LD Player [5] and Sled [42], or the editors CopperAuthor [117] and Reload IMS-LD Editor [5]. As long as the adaptation of the interface is based on the tool and not on the Unit of Learning that is interpreted by the player, this is still true. Today's players do not yet provide facilities to change the size or the position of the navigation panels, or even open and close the working areas in the player. Either, these tools cannot change the style sheets related to a HTML file, part of the content, and any of the linked features, as font-size, font-type or background color, for instance. Although the CopperCore engine provides the appropriate infrastructure, no player uses it so far.

Nevertheless, some kind of adaptive interface is possible, using DIV layers and environments.

Scenario: the options and the look and feel of an interface are adapted on the user's request.

Implementation: regarding activities, several DIV layers or learning activities can be set-up with a different visualization for the same content (for instance, linking the same file to different CSS style sheets). First, we define the different activities:

```
<learning-activity identifier="Activity1InterfaceA">
          <title>Question 1</title>
          <activity-description>
                <item identifierref="firstlessonInterfaceA"/>
          </activity-description>
</learning-activity>

<learning-activity identifier="Activity1InterfaceB">
          <title>Question 1</title>
          <activity-description>
                <item identifierref="firstlessonInterfaceB"/>
          </activity-description>
</learning-activity>
```

Later, we link the CSS style sheets with the same file lesson1.html, resulting in two different resource identifiers:

```
<resource identifier="firstlessonInterfaceA" type="webcontent"
href="lesson1.html">
          <file href="lesson1.html" />
          <file href="stylesheetA.css" />
</resource>

<resource identifier="firstlessonInterfaceB" type="webcontent"
href="lesson1.html">
          <file href="lesson1.html" />
          <file href="stylesheetB.css" />
</resource>
```

And finally, we show and hide the activity linked to the related resource:

```
<if>
    <is>
          <property-ref ref="InterfaceToChoose"/>
          <property-value>A</property-value>
    </is>
</if>
<then>
    <show>
```

```
            <activity-structure-ref ref="firstlessonInterfaceA"/>
      </show>
      <hide>
            <activity-structure-ref ref="firstlessonInterfaceB"/>
      </hide>
  </then>
```

Another possibility could be to adapt not only the look and feel of a DIV layer, but also its content and the options of interaction inside it, resulting in a block of information and interaction.

Last one additional set-up is to define different environments with several contents and services and link them to different activities or activity structures. They will be shown-hidden together with the related activity. This approach can be managed following any of the methods aforementioned. Furthermore, we could count different services, contents and options in every environment that are depending on the learning tree. To some extent, this means a de facto sub-division on the screen and a different adapted interface. So, one activity linked to one environment, and both are shown or hidden, resulting in a final personalised interface:

```
  <show>
      <environment-ref ref="ENVfirstlessonInterfaceA"/>
  </show>
  <hide>
      <environment-ref ref="ENVfirstlessonInterfaceB"/>
  </hide>
```

4.6.4. Review of IMS-LD based on inputs, roles and adaptation

In this section we have shown three main types of adaptation with some typical scenarios and their related implementations, based on the learning flow, the content and the interface. Also, we have stated the four different inputs involved in the adaptation process: a) the student, based on his behavior and his performance; b) the student, based on his personal decision; c) the personal decision of the teacher; and d) the set of rules in an engine, pre-defined by a learning designer. To implement these scenarios in IMS-LD we use the basic structure that Level A provides and the core elements of Level B: properties, global elements, monitoring service, conditions and calculations.

We conclude that it is possible to represent strategies for adaptation taken from all these inputs and types, in IMS-LD. Whether we talk about adaptivity or about adaptability, the issue of personalised learning can be modeled with the specification, using different approaches in order to support learners build better competences and skills.

4.7 . Review of IMS-LD based on specific case studies focused on adaptation

As a core part of this research, in the Annex (Section 10) we define five case studies which implement Units of Learning where all these types of adaptation are depicted. In concrete, these case studies are *Case study, adaptation on the learner´s performance and knowledge: GeoQuiz, Case study, adaptation on the learning designer´s method: QuoBuilder II, Case study, adaptation on the learner´s decision: JazzUK, Case study, adaptation on the teacher´s decision: FSA,* and *Case study, adaptation on the learner´s behaviour: PET*. We use them as a base to find restrictions, drawbacks and elements to improve within the specification. These case studies show how far IMS-LD can go on adaptation, when different inputs and roles are involved. We also make links to the integration of UoLs, when needed. Out of the modeling and development of these Units of Learning we make an analysis on which features, elements and components are missing or could be modified in order to achieve a more adaptive-oriented, expressive general definition, with the ultimate aim of improving the specification and getting it closer to actual and current needs on eLearning. In that Annex, we link the further analysis of this section with the specific feedback right after each case study. Following, we summarize our findings:

- The definition of properties and the link through several working XML files is too complicated to become useful. Even a simple combobox comes to a struggle that has to be defined by hand, as the current editing level of the available editors is too low or too messy
- The relation between layers and actions is not straightforward and it has to be done interlacing files, once more
- The lack of a richer conditional structure makes the editing of the set of rules more complicated on paper than they actually are from a logical point of view
- In case that a user fails and wants to re-fill the questionnaire, how can he/she make it? Once that the questions are answered and the activity is closed (the on completion flag is switched on) the activity cannot be re-initialized and go backwards. Iterations in the activities are not allowed, therefore
- In case that the teacher or the learning designer wants to change the questions, or the answers, or the content of the next activity to carry out, it is not possible to do it, as every single resource has to be packed in design and publishing time before the actual running of the instance. Changes on-the-fly are not possible
- The monitoring service doesn´t cover any kind of user grouping. Therefore, a user cannot follow the performance of several people at the same time

- The teacher-learning designer cannot change the amount of questions or the answers. It is a fixed number for each entry
- Questions and answers are not personalised for every user, and they are the same for everyone in the same role
- The communication between teacher and student is little. They can only see the values of properties but there is no other communication service between them
- There is a lack of flexibility on the input point to raise the question and possibly change the itineraries. For instance, in the type sequence the learning activity with the question is always at the same place. In the type selection, the question is always asked after 2 completed learning activities. What if the learning designer/teacher wants to change the method and shift this input point? From 2 to 3 completed LA´s or from the third in a row to the forth?
- There is no chance to run the Unit of Learning (the whole UoL or a part, such a Learning Activity) two times within the same instance. Once that a Learning Activity is closed, the user can read it again but the associated learning flow cannot be executed. For instance, after that the question to change the itinerary is made in the historisch-route, there is no way to come back
- This UoL is a real course. There is no flexibility to change the content either. When the teacher/learning designer wants to keep the same method and the same structure, but he/she only wants to change one single HTML page with some content, the UoL has to be validated and published again, the learner and the teacher have to be enrolled and the learning process starts from the very beginning
- There is no chance to handle absolute time to start the course and-or a specific activity with. Only relative time to the precise time when the instance is created out of the UoL
- There is no chance to make a connection to an already existing database (for instance, to make a query or to import already enrolled students or teachers). The data type for connection is not supported. Therefore, every enrolment has to be done by hand or running a specific tool for that
- Furthermore, any connection with the external world is impossible (as we addressed in the previous Section on Interoperability). For instance, a real-time effective communication between a LMS and an IMS-LD UoL is no possible, so far, so that they cannot benefit each other from mutual services and resources. There is no dispatcher allowing such this connection

- A file uploaded from the hard disk of the computer of a user is stored in a file-type property inside the internal database of the engine (CopperCore in this case). There is no chance to change the by-default configuration on storing or retrieving resources. There is not any facility to manage such these uploads either. Although this is an issue concerning tools, the core documents of IMS-LD do not provide with this information and-or facility either
- There is a fixed amount of iterations in the reviewing process. The teacher/learning designer can make this amount smaller but he/she cannot make it bigger ever. What happens if the discussion needs one additional round?
- There is no chance no make a dynamic selection of users in order to create groups. The teacher can monitor what happens one by one, and he/she can also provide some feedback to one by one. We could set-up a property to answer by groups, but these groups should be established before the actual start. However, if the teacher wants to make a dynamic creation of a group of students depending on their answers, this is not possible so far
- The connection with an external DataBase, aside from the internal DataBase of the engine (CopperCore 3.0) is not possible so far and it is not in the scope of the specification
- IMS-LD and the database of the engine do not allow for any kind of query from recorded information
- IMS-LD does not allow for recording the behaviour of the user, so that none of measures listed above (i.e., Total Time Needed, Time Before First Move) can be retored or retrieved
- Users cannot be dynamically enrolled within the Unit of Learning, once it has started, and they are managed by an external tool
- IMS-LD doesn´t allow for saving information into external files or for retrieving information from any external source
- We were looking for an example of adaptation based on the user´s behaviour and we found that this cannot be developed with IMS-LD. We also realized that, even when the executable module is developed with other technologies (Macromedia Flash and PHP, for instance) it cannot be integrated with IMS-LD in any way. Therefore, we also found an interoperability problem. Although IMS-LD is not developed thinking of such this kind of interactivity with users, it could allow for a valid integration with external resources using a layer of communication-dispatcher

The main conclusion of this section is that IMS LD does not provide with any specific structure for adaptation, although adaptation can be modelled. The specification will be clearly improved when a set of structures are added to deal with, i.e., content, flow, feedback, and other types of adaptive learning

5. Crítica a la especificación IMS Learning Design desde la investigación tecnológica y educativa

A continuación, unificamos y agrupamos los distintos elementos de análisis y crítica que han surgido en secciones anteriores, basándonos en los estudios sobre integración y adaptación de Unidades de Aprendizaje y que pueden ser encontrados en las secciones anteriores, así como en los anexos (Sección 10), en donde se diseñan, detallan y comentan casos específicos de estudio a través del modelado de Unidades de Aprendizaje. Dividimos los elementos de crítica en tres grupos: Modelado, Arquitectura y Herramienta. Modelado hace referencia a la notación en sí y a las posibilidades de modelado que ofrece mientras que Arquitectura engloba aspectos estructurales y de diseño conceptual de la especificación. Por su parte, Herramienta contempla críticas que, aunque afecta a la divulgación, desarrollo e implementación de IMS-LD, no corresponden a la especificación sino a las herramientas que actualmente existen o a las redes de usuarios implicadas en ellas. Incluimos un identificador único [ID] en cada comentario o crítica del análisis para su posterior seguimiento

5.1 . Modelado

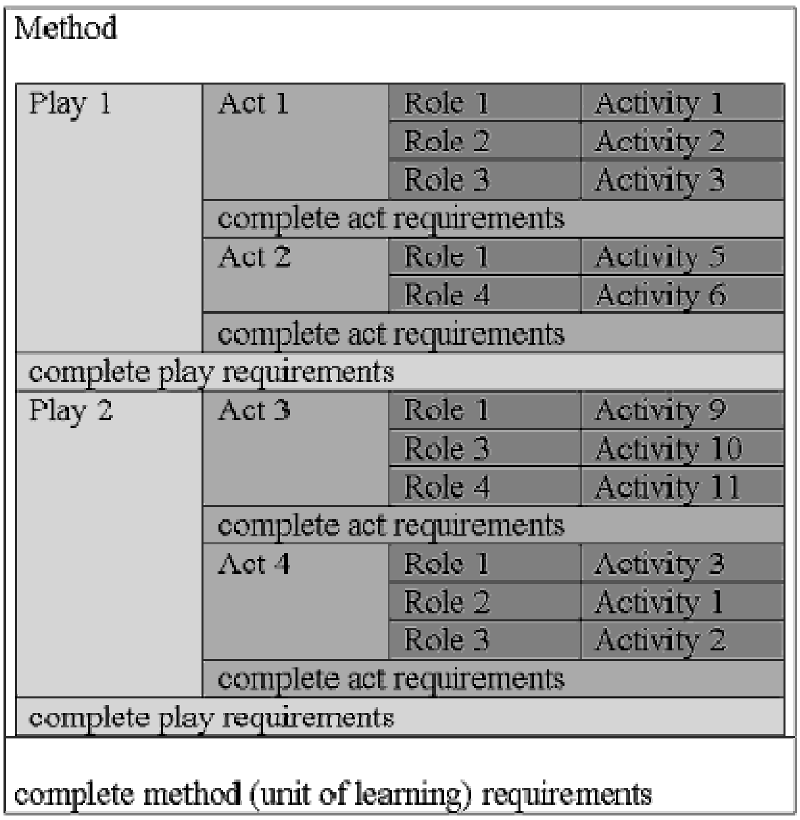


Figure 41. [M.01]

- [M.01] IMS-LD maneja una sintaxis y una gramática incómodas de modelado. Aunque la metáfora que permite definir y relacionar los distintos componentes es clara desde un punto de vista conceptual, no lo es tanto cuando se intenta implementar un ejemplo concreto. La división entre learning design, plays, acts y activity structures, por ejemplo, proporciona una relación en cascada que, siendo versátil para crear escenarios complejos, dista mucho de los escenarios sencillos asociados habitualmente a cursos específicos. Esta subordinación obligatoria de elementos añade confusión

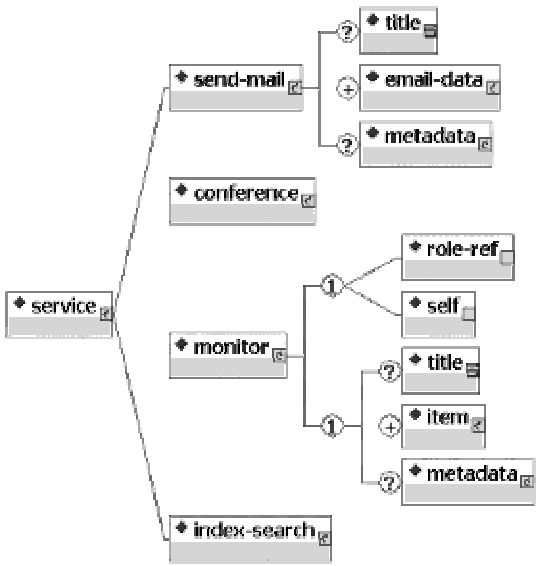


Figure 42. [M.02]

- [M.02] Otro ejemplo sería las referencias consecutivas que definen un servicio y que necesitan definición en la actividad, el entorno, el servicio en sí, el recurso y el fichero que lo ejecuta. Demasiados pasos intermedios para algo tan sencillo

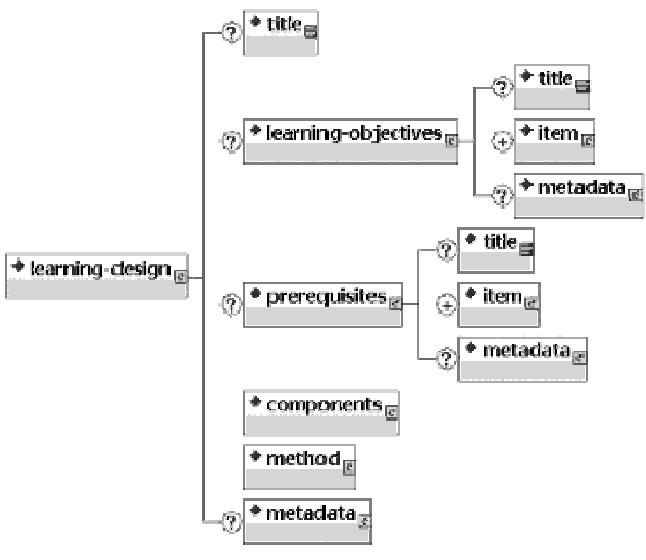


Figure 43. [M.03]

- [M.03] Esta crítica no se extiende al diseño del *learning design* en sí, lógica y portable

- [M.04] En consonancia con el primer comentario, la notación IMS-LD es difícil de concretar. Aunque sigue la filosofía XML, define una estructura propia de manifiesto Content Packaging con una extensión de la etiqueta organizations, que genera un manifiesto Learning Design extenso y farragoso de seguir. Además, las etiquetas y definición de elementos es muy detallada lo que hace laborioso realizar labores sencillas, como una simple definición y uso de variables, por ejemplo. El modelado o programación, por tanto, se vuelven complicados, a pesar de su sencillez conceptual

- [M.05] Las estructuras de programación proporcionadas son muy básicas (condición simple, aritmética simple, edición y visualización de variables, gestión de visibilidad de actividades y capas DIV), además de utilizar una sintaxis extensa, lo que dificulta la visibilidad. Una extensión de la variedad disponible de estas estructuras y una modificación/simplicidad de su sintaxis aumentaría exponencialmente la potencialidad de la especificación en la flexibilidad pedagógica que busca

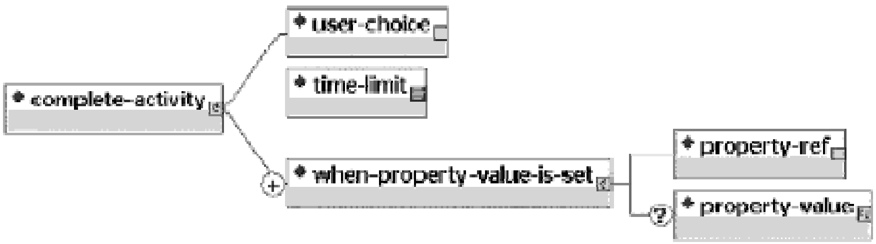


Figure 44. [M.06]

- [M.06] La gestión de tiempo se realiza siempre asociada al comienzo de la instancia de la UoL. No se contempla el tiempo relativo de ejecución desde el momento de comienzo de la ejecución de una actividad por parte del usuario o desde cualquier otro momento de sincronización. Tampoco se contempla la referencia a tiempo absoluto

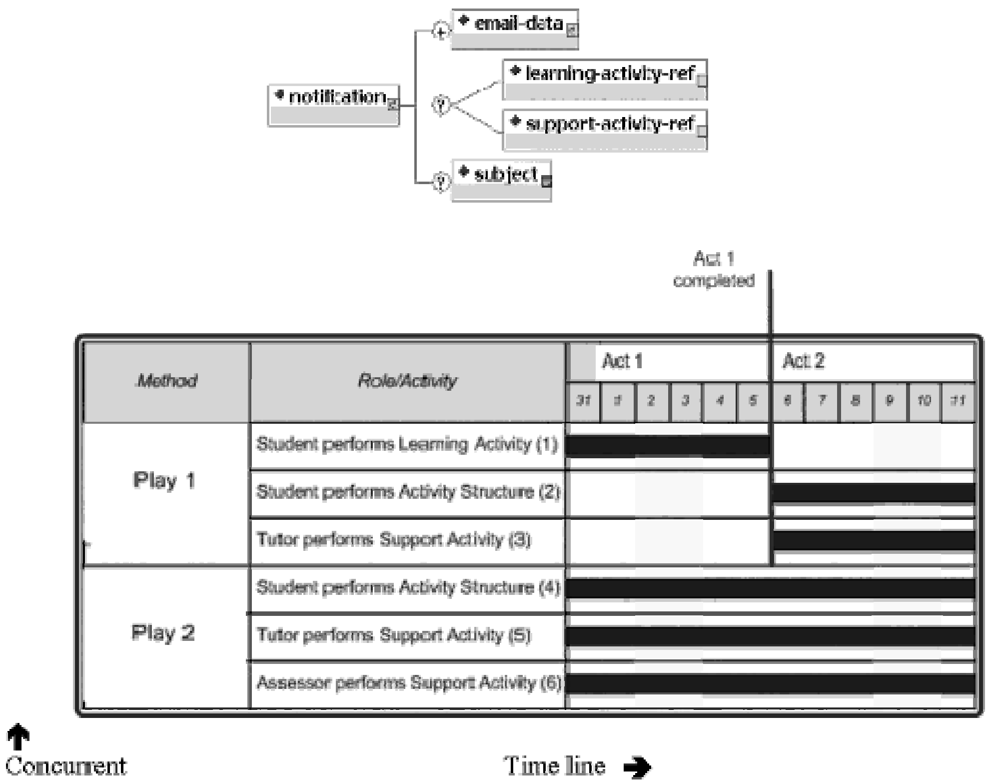


Figure 45. [M.07]

- [M.07] el servicio de notificación del Nivel C contempla únicamente un envío de correo o una realización de una actividad de aprendizaje o soporte, ignorando su concepto básico, como es la ejecución de acciones condicionadas al estado de otros elementos (como la finalización de una actividad de aprendizaje o la satisfacción de una condición en una estructural condicional, por ejemplo). Se encuentra infrautilizado

Desde el punto de vista de la adaptación hemos demostrado que IMS-LD puede llevar a cabo ciertos tipos. Entre ellos, la adaptación del flujo de aprendizaje, del contenido, de la evaluación o de la resolución interactiva de problemas. Parcialmente, puede ejecutar la creación de grupos de usuarios y la adaptación de un curso en run-time, tanto en cuanto hayan sido definidos previamente todos los estados posibles. Por otra parte, esta misma modifiacación en run-time no es posible para la estructura de aprendizaje o para la recuperación dinámica de información. Es cierto, no obstante, que algunas de estas posibilidades no son restricciones estrictas de la especificación, sino del estado actual de las herramientas que la manejan (como la modificación en run-time). En cualquier caso, estos tipos de adaptación son o no realizables dependiendo de los niveles de diseño, interactividad y especificidad que se requieran. Existen limitaciones importantes en función del tipo de modelado, tal y como mostramos a continuación

- [M.08] La definición y uso de propiedades y enlaces entre los diversos ficheros XML que permiten trabajar con los elementos globales son confusos a nivel de estructura en la especificación. La concreción mediante editores técnicos es demasiado laboriosa como para ser ágil. El modelado mediante editores XML es también farragoso. Se necesitan una definición más sencilla y un sistema de enlace y uso más evidente

- [M.09] La relación entre capas DIV y la gestión de la propiedad de visibilidad podría simplificarse para lograr una mayor claridad de definición y seguimiento

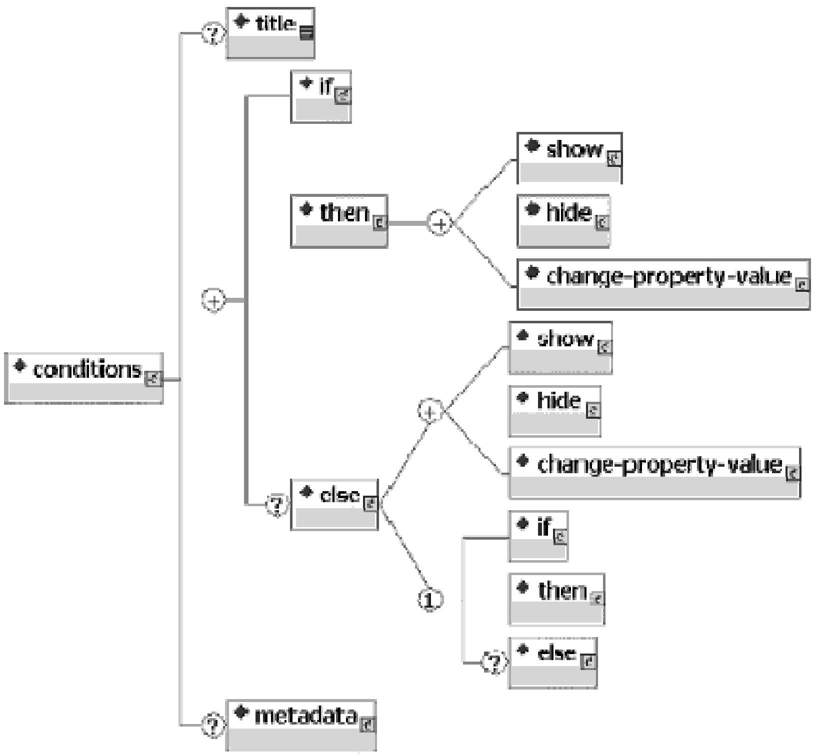


Figure 46. [M.10]

- [M.10] Existe una única manera de establecer condiciones, mediante la estructura if-then-else y anidaciones de la misma. De este modo, la gestión de reglas de comportamiento o cualquier otro tipo de condiciones más complejas se hace innecesariamente complicada

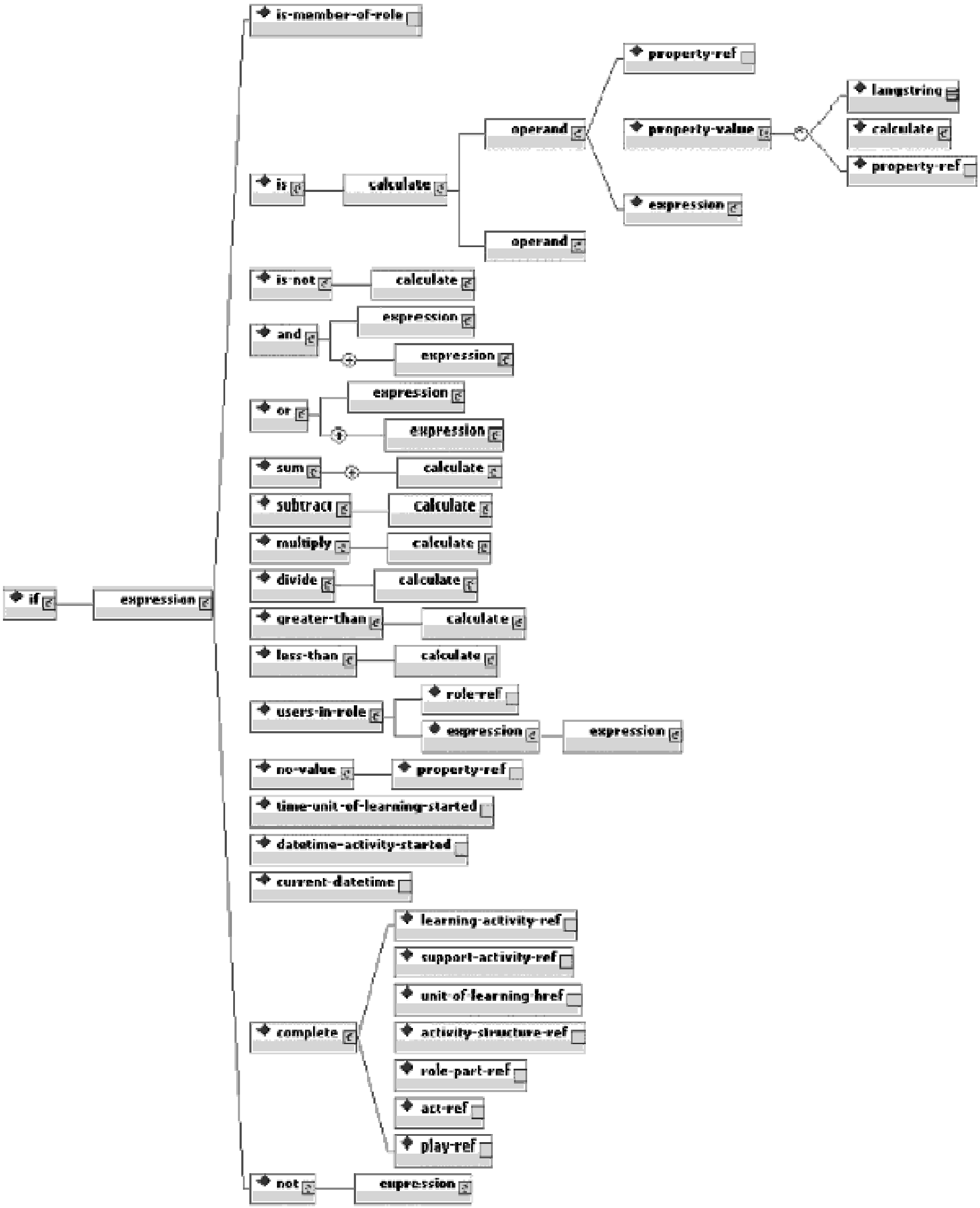


Figure 47. [M.11]

- [M.11] Las expresiones que evalúa la estructura if-then-else son, no obstante, las habituales en este tipo

- [M.12] No existe ninguna estructura de iteración en ninguno de los elementos básicos (actividad, estructura de actividades, actos, obras). Una vez que cualquier de ellos se completa es imposible volver a ejecutarlo con la misma configuración de entrada o alguna otra obtenida como concecuencia de la acción de algún usuario o del método

- [M.13] No existen puntos de entrada o sincronización dentro de la estructura del manifiesto que permitan gestionar el flujo de aprendizaje a voluntad del diseñador de aprendizaje. La predefinición del flujo es demasiado estática

- [M.14] No se puede realizar la asignación concreta de una actividad específica a un usuario seleccionado. Se juega con roles, no con los usuarios de los roles. La excepción que supone la personalización extrema no existe

- [M.15] No existe la creación de grupos de manera selectiva dentro de la instancia. Así, si concurren varios estudiantes que deben ser repartidos en diversos grupos de estudio, no hay forma alguna de hacerlo ni la especificación lo contempla

- [M.16] No existe, por tanto tampoco, una estructura de trabajo colaborativo que no haya sido expresamente modelada con los recursos básicos existentes

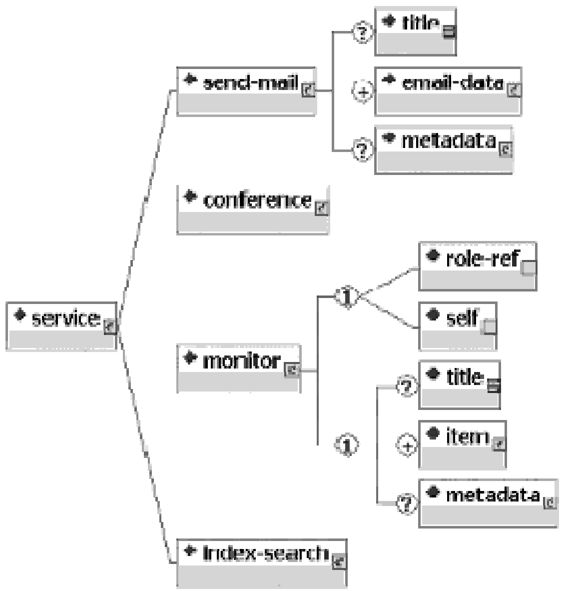


Figure 48. [M.17]

- [M.17] El servicio de monitorización no contempla la monitorización grupal, únicamente la individual, del usuario propio, o de un usuario dentro de un rol. Aunque el rol puede y debe entenderse como un grupo, también demarca una línea de trabajo, impidiendo la agrupación de usuarios seleccionados de roles distintos o la reubicación de usuarios en grupos fuera de sus roles

- [M.18] No existe ningún sistema de comunicación entre usuarios/roles más allá de la monitorización uno a uno mediante la visualización de propiedades ajenas

5.2 . **Arquitectura**

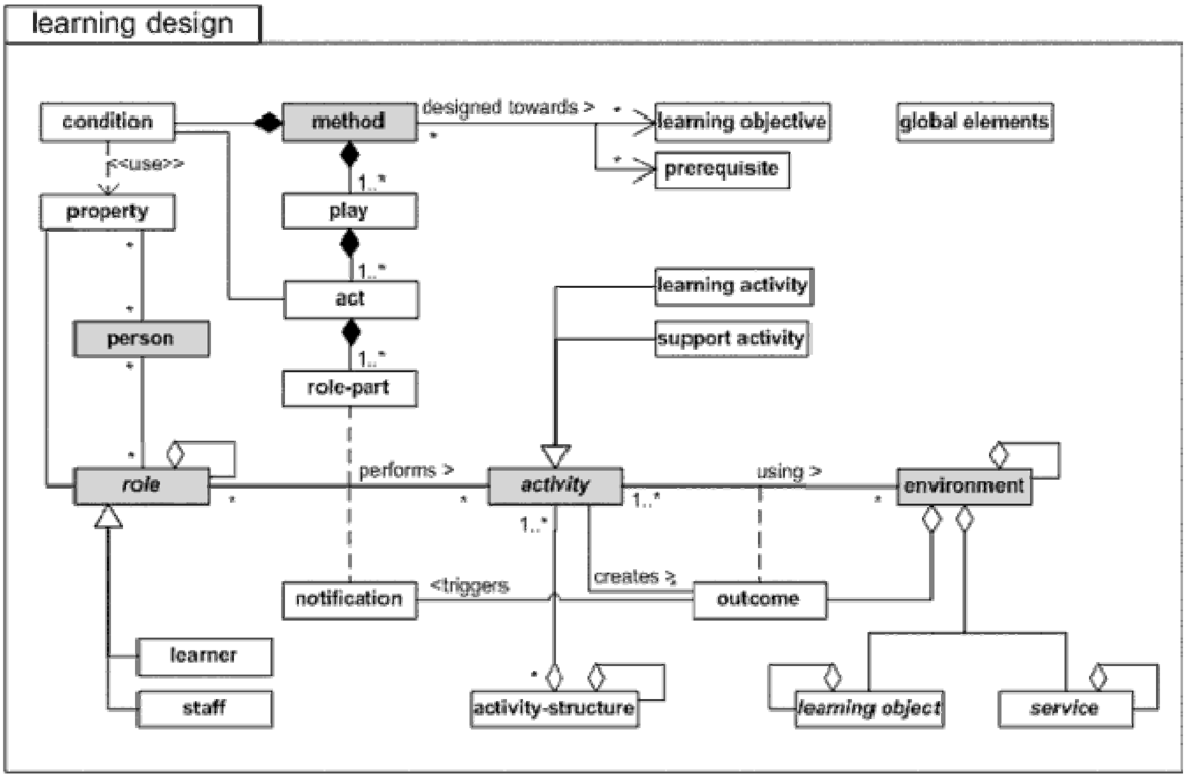


Figure 49. Arquitectura de los tres niveles de IMS-LD

- [A.01] Para lograr la integración con otras especificaciones y módulos externos a IMS-LD es necesario un sistema o capa de comunicación que permita el intercambio efectivo de información, como variables y valores de variables y estado, lo que permitirá interactuar desde IMS-LD en el módulo externo y viceversa, y que tenga en cuenta toda la arquitectura de los tres niveles de la especificación (Figure 49). Esta capa de comunicación debería tener en cuenta las características propias de cada sistema, lenguage, módulo, herramienta o especficación con el que se quiera comunicar, estableciendo unos ficheros externos que detallen la configuración de servicios propios y parámetros propietarios y manteniendo el núcleo fundamental de las estructuturas en un grado de comunicación compatible. La existencia de una capa de comunicación o dispatcher y el desarrollo de la interoperabilidad facilitaría la relación de IMS-LD con módulos y aplicaciones desarrollados con otras especificaciones (IMS Content Packaging, Scorm) otros lenguajes (PHP, Java) y con otros sistemas (Lams, Moodle). Uno de estos grupos lo constituyen los campus virtuales (Learning Management System –LMS- o Virtual Learning Environment –VLE-) desarrollado bajo cánones de IMS-LD o compatible con Unidades de Aprendizaje creadas en IMS-LD, con posibilidad de intercambio (importación y exportación) y ejecución y una capa de servicios (foros, chats, administración de usuarios, seguimiento de expediente, trabajo colaborativo, servicio de noticias o correo) que permita interactuar con ellas

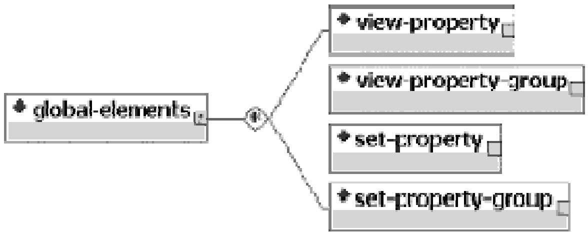


Figure 50. [A.02]

- [A.02] IMS-LD no permite la grabación ni la recuperación de datos en y desde ficheros externos de ningún formato. Así pues, no es posible la importación de información contenida en ficheros de tipo CSV, TXT o RTF, por citar algunos de los formatos más conocidos. A su vez, tampoco es posible el volcado de propiedades o cualquier otro tipo de dato interno en ficheros del mismo tipo. Del mismo modo, las conexiones con bases de datos externas o con módulos programados en otros lenguajes no están contempladas dentro de la especificación. Por ejemplo, cualquier tipo de información contenida en una base de datos estándar SQL (como las fichas de los usuarios) no podrá ser importada, ni exportada, ni consultada mediante un query. Otro ejemplo, la interacción con módulos desarrollados con Macromedia Flash o Java no pasa de la integración mínima como learning objects comunicación con la estructura IMS-LD. La única conexión con sistemas externos se realiza mediante los global elements del Nivel B que permiten visualizar y asignar valores a propiedades o grupos de propiedades en ficheros XHTML, a todas luces insuficiente para explotar las potencialidades de la especificación y para utilizar satisfactoriamente las infinitas posibilidades que otros entornos proporcionan. Se concluye pues que se establece un aislamiento estructural de la especificación con el resto del entorno informático

- [A.03] Existen tres formas de integración de recursos externos con IMS-LD: a) El modelado completo con IMS-LD (lo que no es integración); b) la utilización de IMS-LD como un contenedor de un módulo ajeno, sin comunicación; y c) la plena incorporación del módulo externo dentro del flujo de aprendizaje, con comunicación de variables y estados de manera bidireccional. Mientas que a) depende de lo que se quiera modelar (por ejemplo, en juegos, hemos vista la alta compatibilidad de elementos aislados), y b) se encuentra ampliamente demostrado, dado que reduce el módulo externo a un mero objeto de aprendizaje aislado del itinerario principal, es c) el que representa el reto principal, dado que amplia las posibilidades de una especificación basada en la flexibilidad pedagógica. Estas posibilidades son tantas como recursos, tecnología y metodologías existen y podrán existir o evolucionar en un futuro. Se hace preciso pues una solución genérica, como es el desarrollo de una capa de comunicación o dispatcher que habilite un flujo en ambas direcciones, de IMS-LD al módulo y viceversa, tal y como se ha visto en la Sección 3.11.

- [A.04] Como ya hemos visto, no hay posibilidad de modificar el esqueleto, el método, la definición de roles o cualquier otro elemento estructural en la fase de ejecución o run-time, restringiendo considerablemente las posibilidades del profesor o diseñador de aprendizaje para modificar o adaptar contenidos, itinerario o método a medida que va avanzando el curso y las circunstancias se pueden ir modificando

5.3 . Herramienta

- [H.01] La ejecución actual de las herramientas bajo IMS-LD no soporta la modificación de una Unidad de Aprendizaje en tiempo real, definiendo de facto un divorcio entre el tiempo de diseño y el tiemo de ejecución, con un tiempo de publicación y un tiempo de configuración intermedios. Esta fase de compilación previa a una ejecución encapsulada es habitual en la mayoría de lenguajes actuales (casi todos ellos no intérpretes) pero se enfrenta radicalmente a uno de los conceptos básicos de IMS-LD, como es la adaptación en la autoría y en la ejecución. Del mismo modo, limita profundamente la expresividad pedagógica de la que hace gala IMS-LD dejando al profesor en mero intérprete de un diseño de aprendizaje previo no modificable, ni en estructura ni en contenido, una vez comenzada la ejecución

- [H.02] En referencia a los editores técnicos IMS-LD se necesita una evolución que permita trabajar de manera conjunta con el manifiesto, con los recursos y con los ficheros externos XHTML relacionados, mostrando las dependencias entre ellos y un sistema sencillo de edición y configuración de las propiedades

- [H.03] Juntando la interoperabilidad, con la creación de patrones de aprendizaje y el desarrollo de editores visuales, los profesores y pedagogos podrían modelar Unidades de Aprendizaje centradas en aspectos más específicos, como el aprendizaje personalizado o adaptativo, que constituye uno de los primeros objetivos de IMS-LD [231], el aprendizaje colaborativo, la evaluación, la integración de juegos o el seguimiento de los usuarios en tiempo real [61], por citar algunos

- [H.04] Del mismo modo, junto con la interoperabilidad, el intercambio de paquetes logrará la re-utilización de Unidades de Aprendizaje previamente modeladas, bien para su re-definición, bien para su compilación e inclusión en unidades mayores. Este intercambio de paquetes puede servir de base para el desarrollo de plantillas o patrones de diseños instructivos interoperables, lo que facilitaría la creación y utilización de la especificación por todas las comunidades de práctica involucradas y no únicamente por el sector más tecnológico

- [H.05] Son necesarios editores o herramientas de autoría con un diseño gráfico y de información más centrados en el usuario final y menos en los logros técnicos. Es cierto que este último año hemos pasado de no contar con ninguna aplicación basada en IMS-LD a tener más de una docena centradas o en torno a IMS-LD, lo que supone un gran avance; pero el punto en común de todas ellas es la realización técnica de lo que la especificación define, no la utilización sencilla por usuarios reales. Es decir, con las aplicaciones informáticas actuales se pueden construir Unidades de Aprendizaje, mayormente sencillas o en nivel A, pero el grado de conocimientos técnicos necesarios es alto y la facilidad de uso de los interfaces es escasa. Es necesario un interfaz con un mayor grado de usabilidad, con una metáfora de aplicación y un sistema de diseño gráfico drag&drop, además de una ayuda contextual y bien documentada, en vez de un sistema de solapas y etiquetas para rellenar sin mayor soporte ni información y sin una conexión educativamente metodológica entre ellas, que es lo que existe en la actualidad
- [H.06] Aunque hoy en día únicamente la exportación a otras especificaciones parece posible, incluso para el editor más avanzado (fijándonos por ejemplo en Reload IMS-LD Editor [5; 69] o en Cosmos [45; 70]) la importación y la exportación al 100% representan todavía un obstáculo y constituye por tanto un gran paso para lograr el intercambio efectivo de paquetes de información. De esta manera, se logrará uno de los objetivos de la estandarización del eLearning, como es la interoperabilidad de las unidades o actividades de aprendizaje en diversos sistemas de edición y ejecución, con el objeto de lograr flexibilidad y un mayor grado de autonomía. Esta interoperabilidad es necesaria dentro y fuera de la especificación

Esta última categoría Herramienta realiza un análisis más detallado y complementario de la problemática encontrada en IMS-LD, pero no es objeto de la solución propuesta por esta tesis. En la sección de Anexos se incluye un artículo relacionado con herramientas de autoría que analiza este aspecto (Sección 9.1).

Todos estos elementos de análisis y crítica se muestran en la siguiente tabla Table 12:

Table 12. Resumen de los elementos de análisis y crítica de la especificación IMS-LD

ID	Elementos de análisis y crítica
Modelado	
[M.01]	IMS-LD maneja una sintaxis y una gramática incómodas de modelado. La división entre learning design, plays, acts y activity structures proporciona una subordinación obligatoria de elementos que añade confusión

[M.02]	Las referencias consecutivas que definen un servicio generan demasiados pasos intermedios para algo tan sencillo
[M.03]	El diseño del learning design en sí es lógico y portable
[M.04]	La notación IMS-LD es difícil de concretar. Se genera un manifiesto Learning Design extenso y farragoso de seguir. El modelado o programación, por tanto, se vuelven complicados, a pesar de su sencillez conceptual
[M.05]	Las estructuras de programación proporcionadas son muy básicas (condición simple, aritmética simple, edición y visualización de variables, gestión de visibilidad de actividades y capas DIV)
[M.06]	No se contempla el tiempo relativo de ejecución desde el momento de comienzo de la ejecución de una actividad por parte del usuario o desde cualquier otro momento de sincronización. Tampoco se contempla la referencia a tiempo absoluto
[M.07]	El servicio de notificación del Nivel C contempla únicamente un envío de correo o una realización de una actividad de aprendizaje o soporte. Se encuentra infrautilizado
[M.08]	La definición y uso de propiedades y enlaces entre los diversos ficheros XML que permiten trabajar con los elementos globales son confusos
[M.09]	La relación entre capas DIV y la gestión de la propiedad de visibilidad podría simplificarse para lograr una mayor claridad de definición y seguimiento
[M.10]	Existe una única manera de establecer condiciones, mediante la estructura if-then-else y anidaciones de la misma
[M.11]	Las expresiones que evalúa la estructura if-then-else son, no obstante, las habituales en este tipo de estructuras
[M.12]	No existe ninguna estructura de iteración en ninguno de los elementos básicos (actividad, estructura de actividades, actos, obras)
[M.13]	No existen puntos de entrada o sincronización dentro de la estructura del manifiesto

[M.14]	No se puede realizar la asignación concreta de una actividad específica a un usuario seleccionado
[M.15]	No existe la creación de grupos de manera selectiva dentro de la instancia
[M.16]	No existe una estructura de trabajo colaborativo que no haya sido expresamente modelada con los recursos básicos existentes
[M.17]	El servicio de monitorización no contempla la monitorización grupal
[M.18]	No existe ningún sistema de comunicación entre usuarios/roles más allá de la monitorización uno a uno mediante la visualización de propiedades ajenas
Arquitectura	
[A.01]	Para lograr la interoperabilidad con otras especificaciones y módulos externos a IMS-LD es necesario un sistema o capa de comunicación que permita el intercambio efectivo de información, como variables y valores de variables y estado, lo que permitirá interactuar desde IMS-LD en el módulo externo y viceversa
[A.02]	IMS-LD no permite la grabación ni la recuperación de datos en y desde ficheros externos de ningún formato. Del mismo modo, las conexiones con bases de datos externas o con módulos programados en otros lenguajes no están contempladas dentro de la especificación
[A.03]	Se hace preciso el desarrollo de una capa de comunicación o dispatcher para lograr la integración de módulos externos con IMS-LD
[A.04]	No hay posibilidad de modificar el esqueleto, el método, la definición de roles o cualquier otro elemento estructural en la fase de ejecución o run-time
Herramienta	
[H.01]	La ejecución actual de las herramientas bajo IMS-LD no soporta la modificación de una Unidad de Aprendizaje en tiempo real, definiendo de facto un divorcio entre el tiempo de diseño y el tiemo de ejecución
[H.02]	Se necesita una evolución de los editores técnicos

[H.03]	Se necesita impulsar la creación de patrones de aprendizaje y el desarrollo de editores visuales
[H.04]	El intercambio de paquetes logrará la re-utilización de Unidades de Aprendizaje previamente modeladas, bien para su re-definición, bien para su compilación e inclusión en unidades mayores
[H.05]	Son necesarios editores o herramientas de autoría con un diseño gráfico y de información más centrados en el usuario final y menos en los logros técnicos
[H.06]	La importación y la exportación al 100% representan todavía un obstáculo y constituyen por tanto un gran paso para lograr el intercambio efectivo de paquetes de información

En la siguiente sección abordamos cada uno de los elementos de análisis y crítica detallando la solución propuesta, cuando exista y sea procedente. Al haber distintos puntos relacionados entre sí agrupamos estos en función de la solución que los satisface.

6. **Propuesta de modificación y extensión de IMS-LD**

En esta sección estudiamos cada uno de los elementos de análisis y críticas compilados en la sección anterior y proponemos una serie de modificaciones y extensiones de la especificación IMS-LD que hacen referencia tanto a Modelado como a Arquitectura. El apartado sobre Herramientas que apuntábamos anteriormente se encuentra fuera del alcance de esta tesis, en tanto en cuanto no afecta a la especificación en sí, sino a la manera en que esta es interpretada por la tecnología y los desarrolladores del momento. Nos centramos pues en la modificación estructural de la especificación y no en su concreción en herramientas concretas. Aun así, dada la relevancia que para la especificación tienen los aspectos de autoría y divulgación, en su momento incorporamos dos estudios relacionados con editores visuales y con definición de plantillas y escenarios de aprendizaje.

En referencia a las propuestas concretas sobre sintaxis y gramática se busca la máxima integración y la máxima funcionalidad con el mínimo impacto dentro de la estructura posible, es decir, respetar al máximo la especificación tal y como es con sus definiciones actuales y realizar los cambios e incorporaciones mínimos necesarios para completarla y extenderla acorde con nuestro análisis. Para mayor claridad, y como hasta ahora en cada fragmento de código, se ha eliminado el prefijo `IMS-LD:` de toda etiqueta.

ID	Elemento de análisis y/o crítica
Modelado	
[M.05] [M.10] [M.11]	Las estructuras de programación proporcionadas son muy básicas (condición simple, aritmética simple, edición y uso de variables, gestión de visibilidad de actividades y capas DIV)

Modificación y/o extensión propuestas	
[Ext.01a]	Estructura condicional de tipo case
Descripción	
Permite las acciones múltiples distribuidas por rangos de cumplimiento de una serie de condiciones por parte de la misma propiedad. Sin limitación de número <case-conditions> representa el grupo de evaluaciones (greater-than, less-than, is…) que realiza la estructura case para ejecutar las acciones asociadas, incluyendo una acción alternativa `else`. Puede compararse con propiedades (`ID-PROPERTY`) o con valores concretos (`VALUE`) <action> representa el conjunto formado por x=1..n acciones individuales a ejecutar, del tipo condición, notificación, cambio de propiedad, cálculo… representadas a su vez por [Ax]	
Sintaxis	

```
<case>
      <property-ref ref="ID-PROPERTY"/>
</case>

<case-conditions>
      <greater-than> <!-- greater-than, less-than, is -->
```

```
             <property-ref ref="ID-PROPERTY"/>
      </greater-than>
            <action>
              [A1..n] <!-- show/hide,change-property,notification...-->
            </action>

      <less-than>
             <property-value>VALUE</property-value>
      </less-than>
            <action>
              [An+1..m]
            </action>

      <!-- ... -->

      <else>
            <action>
              [Am+1..p]
            </action>
      </else>
</case-conditions>
```

Ejemplo *(en cursiva, declaraciones relacionadas; en **negrita**, la modificación/extensión propuesta)*

```
<!-- ... -->

<loc-property identifier="score">
      <datatype datatype="integer"/>
      <initial-value>0</initial-value>
</loc-property>

<loc-property identifier="threshold">
      <datatype datatype="integer"/>
      <initial-value>5</initial-value>
</loc-property>

<!-- ... -->

<learning-activity identifier="passed">
      <title>You passed</title>
      <activity-description>
        <item identifierref="res-passed" identifier="I-passed" />
      </activity-description>
</learning-activity>

<learning-activity identifier="not-passed">
      <title>You did not passed</title>
      <activity-description>
        <item identifierref="res-not-passed" identifier="I-not-passed"
/>
      </activity-description>
</learning-activity>

<!-- ... -->

<case>
      <property-ref ref="score"/>
</case>

<case-conditions>
            <less-than>
                   <property-ref ref="threshold"/>
            </less-than>
```

```
                  <action>
                    <show>
                       <learning-activity-ref ref="not-passed" />
                    </show>
                  </action>
            <greater-than>
                   <property-value>VALUE</property-value>
            </greater-than>
                  <action>
                    <show>
                        <learning-activity-ref ref="passed" />
                    </show>
                  </action>
            <else>

<!-- You are right on the threshold, so you get both activities -->

                  <action>
                    <show>
                      <learning-activity-ref ref="not-passed" />
                      <learning-activity-ref ref="passed" />
                    </show>
                  </action>
            </else>
</case-conditions>
```

Modificación y/o extensión propuestas	
[Ext.01b]	Estructura condicional de tipo case con rangos automáticos
Descripción	
Variante de la estructura case propuesta en [Ext.01a] donde las condiciones a cumplir por la propiedad de entrada se establecen por rangos numéricos complementarios que se definen de manera automática Los rangos abarcan: Mayor o igual que ANTERIOR, menor que ACTUAL. Excepto para el último umbral: Mayor o igual que ANTERIOR, menor o igual que ULTIMO Especialmente útil para evaluación de cuestionarios Los distintos umbrales de los rangos pueden estar definidos como valores directos o como valores almacenados en propiedades, indistintamente No admite alternativa <else> al ser considerada esta estructura case una gestión cerrada de uso específico	
Sintaxis	

```
<case-threshold>
      <property-ref ref="ID-PROPERTY"/>
</case-threshold>

<case-conditions>
      <threshold>
             <property-value>VALUE-BOTTOM</property-value>
      </threshold>
      <threshold>
            <property-ref ref="ID-PROPERTY-1"/>
             <!-- ge VALUE-BOTTOM AND lt ID-PROPERTY-1 -->
            <action>
              [A1..n]
            </action>
      </threshold>
      <threshold>
      <threshold>
            <property-ref ref="ID-PROPERTY-2"/>
             <!-- ge ID-PROPERTY-1 AND lt ID-PROPERTY-2 -->
            <action>
```

```
            [An+1..m]
          </action>
    </threshold>

    <!-- ...with X properties and-or values -->

    <threshold>
          <property-value>VALUE-TOP</property-value>
           <!-- ge ID-PROPERTY-X AND le VALUE-TOP -->
          <action>
            [Am+1..p]
          </action>
    </threshold>
</case-conditions>
```

Ejemplo *(en cursiva, declaraciones relacionadas; en **negrita**, la modificación/extensión propuesta)*

```
<!-- ... -->

<locpers-property identifier="prop-score">
      <datatype datatype="integer"/>
</locpers-property>

<loc-property identifier="prop-A">
      <datatype datatype="integer"/>
      <initial-value>100</initial-value>
</loc-property>

<loc-property identifier="prop-C">
      <datatype datatype="integer"/>
      <initial-value>50</initial-value>
</loc-property>

<!-- ... -->

<learning-activity identifier="A-grade">
      <title>Excellent</title>
      <activity-description>
        <item identifierref="res-A-grade" identifier="I-A-grade" />
      </activity-description>
</learning-activity>

<learning-activity identifier="B-grade">
      <title>Very good</title>
      <activity-description>
        <item identifierref="res-B-grade" identifier="I-B-grade" />
      </activity-description>
</learning-activity>

<!-- ... -->

<case-threshold>
      <property-ref ref="prop-score"/>
</case-threshold>

<case-conditions>
      <threshold>
             <property-value>0</property-value>
      </threshold>
      <threshold>
            <property-ref ref="prop-C"/>
             <!-- [0-49] -->
            <action>
```

```
            <!-- notification to the teacher, as defined later on -->
        </action>
    </threshold>
    <threshold>
    <threshold>
         <property-value>75</property-value>
         <!-- [50-74] -->
        <action>
            <learning-activity-ref ref="B-grade" />
        </action>
    </threshold>
    <threshold>
        <property-ref ref="prop-A"/>
         <!-- [75-100] -->
        <action>
            <learning-activity-ref ref="A-grade" />
        </action>
    </threshold>
</case-conditions>
```

Modificación y/o extensión propuestas	
[Ext.02]	Bucle de condición inicial tipo while

Descripción

Con la condición de entrada y continuación de bucle las iteraciones se siguen produciendo hasta que deje de cumplirse la misma
[condition] representa cualquiera de las condiciones actuales de IMS-LD (greater-than, less-than, is…)
<action> representa el conjunto formado por x=1..n acciones individuales a ejecutar, del tipo condición, notificación, cambio de propiedad, cálculo… representadas a su vez por [Ax]
Se incorpora comando break para forzar la terminación de las iteraciones y salto a la siguiente estructura que corresponda por flujo

Sintaxis

```
<when>
    [condition]
    <action>
        [A1]
        [A2]
        ...
        <break>
        ...
        [An]
    </action>
</when>
```

Ejemplo

```
<!-- ... -->

<locpers-property identifier="end-intro">
    <datatype datatype="boolean"/>
    <initial-value>1</initial-value>
</locpers-property>

<locpers-property identifier="state">
    <datatype datatype="string"/>
    <initial-value>Welcome to the quiz!</initial-value>
</locpers-property>

<!-- ... -->
```

```
<learning-activity identifier="quiz">
      <title>How much do you know?</title>
      <activity-description>
            <item identifierref="res-quiz" identifier="I-quiz" />
      </activity-description>
      <complete-activity>
            <user-choice/>
      </complete-activity>
</learning-activity>

<!-- ... -->

<when>
      <is-not>
             <property-ref ref="end-intro"/>
             <property-value>1</property-value>
      </is-not>
      <action>
            <show>
                  <learning-activity-ref ref="quiz" />
            </show>
            <change-property-value>
                  <property-ref ref="state"/>
                  <property-value>Still in!</property-value>
            </change-property-value>
            <notification>
                  <!-- ... -->
            </notification>
      </action>
</when>
```

Modificación y/o extensión propuestas	
[Ext.03]	Bucle tipo for-next

Descripción

Se realizan tantas iteraciones como indique la variable de umbral que puede compararse con propiedades (ID-PROPERTY) o con valores concretos (VALUE)
<action> representa el conjunto formado por x=1..n acciones individuales a ejecutar, del tipo condición, notificación, cambio de propiedad, cálculo… representadas a su vez por [Ax]
ID-ITER modifica su valor a cada iteración por el valor del contador de la misma, comenzando por el valor configurado por defecto. De esta manera, si VALUE almacena 3 e ID-ITER tiene un valor por defecto de 5, ID-ITER contendrá sucesivamente 5, 6 y 7
Se incorpora comando break para forzar la terminación de las iteraciones y salto a la siguiente estructura que corresponda por flujo

Sintaxis

```
<for>
      <is>
             <property-ref ref="ID-ITER"/>
             <property-value>VALUE</property-value>
      </is>
      <action>
            [A1]
            [A2]
            ...
            <break>
            ...
            [An]
      </action>
```

```
</for>

<!-- OR -->

<for>
      <is>
            <property-ref ref="ID-ITER"/>
            <property-ref ref="ID-PROPERTY"/>
      </is>
      <action>
            [A1]
            [A2]
            ...
            <break>
            ...
            [An]
      </action>
</for>
```

Ejemplo

```
<!-- ... -->

<locpers-property identifier="ID-ITER">
      <datatype datatype="integer"/>
      <initial-value>0</initial-value>
</locpers-property>

<locpers-property identifier="ID-MAX">
      <datatype datatype="integer"/>
      <initial-value>3</initial-value>
</locpers-property>

<locpers-property identifier="state">
      <datatype datatype="string"/>
      <initial-value>Welcome to the quiz!</initial-value>
</locpers-property>

<!-- ... -->

<learning-activity identifier="quiz">
      <title>How much do you know?</title>
      <activity-description>
            <item identifierref="res-quiz" identifier="I-quiz" />
      </activity-description>
      <complete-activity>
            <user-choice/>
      </complete-activity>
</learning-activity>

<!-- ... -->

<for>
      <is>
             <property-ref ref="ID-ITER"/>
             <property-ref ref="ID-MAX"/>
      </is>
      <action>
            <show>
                  <learning-activity-ref ref="quiz" />
            </show>
            <change-property-value>
                  <property-ref ref="state"/>
```

```
                    <property-ref ref="ID-ITER"/>
            </change-property-value>
            <notification>
                  <!-- ... -->
            </notification>
      </action>
</when>
```

Modificación y/o extensión propuestas	
[Ext.04]	Modificación de la estructura de cálculo <calculate>
Descripción	
Permite realizar operaciones combinadas de aritmética simple (adición, sustracción, división y multiplicación) de manera secuencial, y no anidada Permite la aplicación del mismo operador a una sequencia de operandos Permite agrupar operaciones Incorpora el cálculo de porcentaje y media, como ejemplos de extensión	
Sintaxis	

```
<!-- operatorX could be sum, subtract, multiply, division -->
<!-- operatorX could also be average, percent, but with their specific
sintax -->

<calculate>
<!-- "ID-PROPERTY" operator1 VALUE operator2 "ID-PROPERTY-2" -->
      <property-ref ref="ID-PROPERTY" />
      <operator1>
            <property-value>VALUE</property-value>
      </operator1>
      <operator2>
            <property-ref ref="ID-PROPERTY-2" />
      </operator2>
</calculate>

<calculate>
<!-- "ID-PROPERTY" operator1 VALUE operator1 "ID-PROPERTY-2" -->
      <operator1>
            <property-ref ref="ID-PROPERTY" />
            <property-value>VALUE</property-value>
            <property-ref ref="ID-PROPERTY-2" />
      </operator1>
</calculate>

<calculate>
<!-- "ID-PROPERTY" operator2 ("ID-PROPERTY-2" operator1 VALUE) -->
      <property-ref ref="ID-PROPERTY-2" />
      <operator2>
      <group-subtotal ref="SUB-1">
            <property-ref ref="ID-PROPERTY" />
            <operator1>
                  <property-value>VALUE</property-value>
            </operator1>
      </group-subtotal>
      </operator2>
</calculate>

<calculate>
<!-- AVERAGE of ("ID-PROPERTY" + VALUE + "ID-PROPERTY-2") -->
      <average>
            <property-ref ref="ID-PROPERTY" />
            <property-value>VALUE</property-value>
            <property-ref ref="ID-PROPERTY-2" />
```

```
      </average>
</calculate>

<calculate>
<!-- "ID-PERCENTAGE" of "ID-BASE" -->
      <percent>
            <property-ref ref="ID-BASE" />
            <property-ref ref="ID-PERCENTAGE" />
      </percent>
</calculate>
```

Ejemplo

```
<!-- ... -->

<locpers-property identifier="ID-OP-1">
      <datatype datatype="integer"/>
      <initial-value>7</initial-value>
</locpers-property>

<locpers-property identifier="ID-OP-2">
      <datatype datatype="integer"/>
      <initial-value>5</initial-value>
</locpers-property>

<locpers-property identifier="ID-OP-3">
      <datatype datatype="real"/>
      <initial-value>54.0</initial-value>
</locpers-property>

<locpers-property identifier="ID-PERCENTAGE">
      <datatype datatype="real"/>
      <initial-value>20.0</initial-value>
</locpers-property>

<!-- ... -->

<calculate>
<!-- "ID-OP-1" + 3 - "ID-OP-2" -->
      <property-ref ref="ID-OP-1" />
      <sum>
            <property-value>3</property-value>
      </sum>
      <subtract>
            <property-ref ref="ID-OP-2" />
      </subtract>
</calculate>

<calculate>
<!-- "ID-OP-1" + 3 + "ID-OP-2" -->
      <sum>
            <property-ref ref="ID-OP-1" />
            <property-value>3</property-value>
            <property-ref ref="ID-OP-2" />
      </sum>
</calculate>

<calculate>
<!-- "ID-OP-1" * ("ID-OP-2" + 3) -->
      <property-ref ref="ID-OP-1" />
      <multiply>
      <group-subtotal ref="SUB-1">
            <property-ref ref="ID-OP-2" />
```

```
            <sum>
                  <property-value>3</property-value>
            </sum>
      </group-subtotal>
      </multiply>
</calculate>

<calculate>
<!-- ("ID-OP-1" + 3 + "ID-OP-2") / 3 -->
      <average>
            <property-ref ref="ID-OP-1" />
            <property-value>3</property-value>
            <property-ref ref="ID-OP-2" />
      </average>
</calculate>

<calculate>
<!-- "ID-PERCENTAGE" * "ID-OP-3" / 100 -->
      <percent>
            <property-ref ref="ID-OP-3" />
            <property-ref ref="ID-PERCENTAGE" />
      </percent>
</calculate>
```

ID	Elemento de análisis y/o crítica
Modelado	
[M.06]	No se contempla el tiempo relativo de ejecución desde el momento de comienzo de la ejecución de una actividad por parte del usuario o desde cualquier otro momento de sincronización. Tampoco se contempla la referencia a tiempo absoluto

Modificación y/o extensión propuestas	
[Ext.05]	Se modifica la referencia al tiempo relativo y se añade la referencia a tiempo absoluto

Descripción

Se modifica la gestión de tiempo relativo para incluir el momento de inicio por parte de un usuario de un acto, una obra, una actividad de aprendizaje, una actividad de soporte, una notificacion, una propiedad activa que actúa a modo de disparador o flag, un role-part o un timestamp almacenado en una propiedad de tipo string, configurable desde la propia instancia en ejecución. Todos compatibles con la indicación de tiempo relativo al comienzo de la instancia (run) que puede ser almacenado en una propiedad de tipo string

Se añade la referencia a tiempo absoluto

Sintaxis

```
<!-- Cada entrada <time-limit> en una uol, un play, un acto o una
actividad puede contemplar la inclusión de una referencia a un momento
concreto -->

<!-- Así, dentro de las partes de finalización de cada una de las
siguientes estructuras: -->

<complete-unit-of-learning>
<!-- in 2 months -->
      <time-limit>P2M</time-limit>
</complete-unit-of-learning>

<complete-play>
<!-- in 1 month and 25 days -->
      <time-limit>P1M25D</time-limit>
```

```
</complete-play>

<complete-act>
<!-- in 1 month and 20 days -->
      <time-limit>P1M20D</time-limit>
</complete-act>

<complete-activity>
<!-- in 1 month and 15 days -->
      <time-limit>P1M15D</time-limit>
</complete-activity>

<!-- pasaríamos a sustituir <time-limit>P1M15D</time-limit> por: -->

<time-limit>P1M15D</time-limit> <!-- referencia actual a la run -->

<!-- or -->

<time-limit>P1M15D</time-limit>
<act-ref ref="ID-ACT"/ >

<!-- or -->

<time-limit>P1M15D</time-limit>
<play-ref ref="ID-PLAY"/ >

<!-- or -->

<time-limit>P1M15D</time-limit>
<learning-activity-ref ref="ID-LA"/ >

<!-- or -->

<time-limit>P1M15D</time-limit>
<role-part-ref ref="ID-ROLE-PART"/ >

<!-- or -->

<time-limit>P1M15D</time-limit>
<support-activity-ref ref="ID-SA"/ >

<!-- or -->

<time-limit>P1M15D</time-limit>
<notification ref="ID-NOTIFICACION"/ >

<!-- or -->

<time-limit>P1M15D</time-limit>
<property-ref ref="ID-PROPERTY-FLAG"/ > <!-- type boolean -->

<!-- or -->

<time-limit>P1M15D</time-limit>
<property-ref ref="ID-PROPERTY-TIMESTAMP"/ > <!-- type string -->

<!-- Del mismo modo,se puede incluir la referencia a tiempo absoluto -->

<time-limit-abs>P2007Y2M17D</time-limit-abs> <!-- February 17th,2007 -->
```

Ejemplo

```
<!-- ... -->

<locpers-property identifier="ID-PROPERTY-TIMESTAMP">
      <title>Timestamp</title>
      <datatype datatype="string"/>
      <initial-value>P2007Y2M17D9H</initial-value>
            <!-- February 17th, 2007, 9AM -->
</locpers-property>

<!-- ... -->

<learning-activity identifier="ID-LA-previous" isvisible="true">
      <title>AboutRelativeTime</title>
      <activity-description>
            <title>CheckingTime</title>
            <item identifier="item-LA-previous"
                  identifierref="res-LA-previous" isvisible="true" />
      </activity-description>
      <complete-activity>
            <user-choice/>
      </complete-activity>
</learning-activity>

<learning-activity identifier="ID-LA-1" isvisible="true">
      <title>AboutRelativeTime</title>
      <activity-description>
            <title>CheckingTime</title>
            <item identifier="item-LA-1"
                  identifierref="res-LA-1" isvisible="true" />
      </activity-description>
      <complete-activity>
            <time-limit>P1M15D</time-limit>
                  <!-- 1 month, 15 days,from end of "ID-LA-previous" -->
            <learning-activity-ref ref="ID-LA-previous"/ >
      </complete-activity>
</learning-activity>

<!-- ... -->

<act identifier="act-1">
      <title>Act</title>
            <role-part identifier="rolepart-1">
                  <title>Role Part</title>
                  <role-ref ref="role-student" />
                  <learning-activity-ref ref="ID-LA-previous" />
            </role-part>
            <complete-act>
                  <time-limit>P10D</time-limit>
                        <!-- 10 days from "ID-PROPERTY-TIMESTAMP" -->
                  <property-ref ref="ID-PROPERTY-TIMESTAMP"/ >
            </complete-act>
</act>
```

ID	Elemento de análisis y/o crítica
Modelado	
[M.07]	El servicio de notificación del Nivel C contempla únicamente un envío de correo o una realización de una actividad de aprendizaje o soporte. Se encuentra infrautilizada

Modificación y/o extensión propuestas

[Ext.06]	Extensión del servicio de notificación para que admita otras acciones aparte de sendmail y actividad y para que pueda ser llamado desde otras estructuras y no únicamente desde la parte <on-completion> de una learning activity

Descripción

Se amplia la lista de elementos sostenibles por el servicio de notificación del nivel C abarcando: `sendmail, learning activity, support activity, condition`
También se amplían los servicios que describiremos más adelante: import, export, set-db, get-db y otros, reforzando el concepto de *trigger* que sustenta la definición de notificación
Pueden existir estos servicios de 1 a n ejecutados en secuencia
El servicio de notificación no se encuentra exclusivamente sujeto a la condición de finalización de una learning activity sino que se instituye como una acción que puede ser lanzada como parte de una condición, de una estructura case, de una estructura de iteración, etc.

Sintaxis

```
<!-- ...  -->

<!-- Setting-up the receiver  -->
<globpers-property identifier="prop-email">
      <global-definition uri="http://whoknows.nl">
        <title>Email address</title>
        <datatype datatype="string" />
        <initial-value>daniel.burgos@ou.nl</initial-value>
      </global-definition>
</globpers-property>

<!-- ...  -->

<!-- Invisible activity  to be activated when the notification starts-->
<learning-activity identifier="la-triggered" isvisible="false">
      <title>Second activity</title>
      <activity-description>
            <title>Activity to be launched</title>
            <item identifier="item-activity-to"
                identifierref="resource-activity-to" isvisible="true" />
      </activity-description>
      <complete-activity>
            <user-choice />
      </complete-activity>
</learning-activity>

<!-- Mother activity, before the notification -->
<learning-activity identifier="la-email" isvisible="true">
      <title>First activity</title>
      <activity-description>
        <title>Activity to launch the notif. from</title>
            <item identifier="item-activity-from"
              identifierref="resource-activity-from" isvisible="true" />
      </activity-description>
      <complete-activity>
            <user-choice />
      </complete-activity>
      <on-completion>

<!-- When the mother activity is finished, the notification starts -->
        <notification>
         <email-data email-property-ref="prop-email">
           <role-ref ref="role-learner" />
           <subject>This is the triggered activity</subject>
```

```
        </email-data>

                <!-- and-or -->

                <learning-activity-ref ref="la-triggered" />

                <!-- and-or -->

                <export-ref ref="export" />

                <!-- and-or -->

                <set-db-ref ref="set-db" />

                <!-- and-or -->

                <get-db-ref ref="get-db" />

                <!-- and-or -->

                <import-ref ref="import" />

                <!-- and-or -->

                <if>
                      <!-- ... -->
                </if>
            </notification>
      </on-completion>
</learning-activity>

<!-- ... -->
```

Ejemplo

```
<!-- ...  -->

<globpers-property identifier="prop-email">
      <global-definition uri="http://www.cubanrevolution.org">
            <title>Email address</title>
            <datatype datatype="string" />
            <initial-value>tutor@university.edu</initial-value>
      </global-definition>
</globpers-property>

<!-- ...  -->

<learning-activity identifier="la-help" isvisible="false">
      <title>Help for advanced users</title>
      <activity-description>
            <title>Help with the advanced course</title>
            <item identifier="item-page1" identifierref="resource-page1"
isvisible="true" />
      </activity-description>
      <complete-activity>
            <user-choice />
      </complete-activity>
</learning-activity>

<learning-activity identifier="la-email" isvisible="true">
      <title>First discussion</title>
      <activity-description>
            <title>What is the future of the Cuban revolution?</title>
```

```
            <item identifier="item-fidel" identifierref="resource-fidel"
isvisible="true" />
      </activity-description>
      <complete-activity>
            <user-choice />
      </complete-activity>
      <on-completion>
        <notification>
          <learning-activity-ref ref="la-help" />
          <export-ref ref="export" />
          <email-data email-property-ref="prop-email">
            <role-ref ref="role-learner" />
            <subject>Help started. Back-up ready</subject>
          </email-data>
        </notification>
      </on-completion>
</learning-activity>

<!-- ... -->
```

ID	Elemento de análisis y/o crítica
Modelado	
[M.08]	La definición y uso de propiedades y enlaces entre los diversos ficheros XML que permiten trabajar con los elementos globales son confusos

Modificación y/o extensión propuestas	
[Ext.07]	Modificación de la sintaxis, definición y utilización de los elementos view-property y set-property, así como de la definición de las variables que los utilizan

Descripción

Se asigna un identificador único a cada variable o elemento que sea susceptible de ser utilizado como elemento global en un fichero XHTML externo (ver [Ext.22], [Ext.23], [Ext.24], [Ext.25]) independientemente de la naturaleza del mismo
Se sustituyen view-property y set-property por los elementos view y set, pudiéndose referir a cualquier identificador único válido (por ejemplo, propiedades)
Se amplía la configuración de cada variable indicando los ficheros que hacen uso de ella. De esta manera, se establece un índice que facilita la relación entre los ficheros y la identificación de las propiedades
Por defecto, al utilizar el identificador se entiende como referido a propio (self). Si se refiere a otros, se amplia la notación a *supported-person*
Se simplifica la notación para visualizar y asignar valores a propiedades desde ficheros externos a través de los globalelementes
Por defecto, se visualiza o se asigna el valor de la variable (value), luego no hace falta indicarlo

Sintaxis

```
<!-- in the MANIFEST -->

<locpers-property identifier="Name">
      <datatype datatype="string"/>
      <initial-value>Martin</initial-value>
      <files>file1..n</files>
</locpers-property>

<!-- in the external XHTML files -->

<!-- from -->
  <set-property xmlns="http://www.imsglobal.org/xsd/IMS-LD_v1p0"
ref="Name" property-of="self" view="value"/>
```

```
<!-- to -->
  <set ref="Name"/>

<!-- from -->
  <view-property ref="Name" property-of="self" view="value"/>
<!-- to -->
  <view ref="Name"/>

<!-- from -->
<view-property xmlns="http://www.imsglobal.org/xsd/IMS-LD_v1p0"
ref="Name" property-of="supported-person" view="value"/>
<!-- to -->
<set ref="Name" property-of="supported-person"/>
```

Ejemplo

```
<!-- in the file MANIFEST -->

<locpers-property identifier="Name">
      <datatype datatype="string"/>
      <initial-value>Martin</initial-value>
      <files>file-question, file-answer</files>
</locpers-property>

<!-- in the external XHTML file: file-question -->

<html xmlns:IMS-LD="http://www.imsglobal.org/xsd/IMS-LD_v1p0"
xmlns="http://www.w3.org/1999/xhtml">

Please, enter your name: <set ref="Name"/>

<!-- in the external XHTML file: file-answer -->

<html xmlns:IMS-LD="http://www.imsglobal.org/xsd/IMS-LD_v1p0"
xmlns="http://www.w3.org/1999/xhtml">

Dear <view ref="Name"/>, this is your answer:
```

ID	Elemento de análisis y/o crítica
Modelado	
[M.09]	La relación entre capas DIV y la gestión de la propiedad de visibilidad podría simplificarse para lograr una mayor claridad de definición y seguimiento

Modificación y/o extensión propuestas	
[Ext.08]	Por defecto, asumir como estado hide la propiedad de visibilidad de capas para conseguir que la estructura condicional que las gestiona se simplifique

Descripción

Las capas comienzan ocultas, que es su estado por defecto. Únicamente se visualizan cuando se activan expresamente con el comando show. Después de la primera aparición se gestionan con show y hide como actualmente
En el ejemplo siguiente con dos capas, se reduce el volumen de código en un 40%

Sintaxis

```
<!-- La sintaxis original no sufre modificaciones aunque sí el modelado
asociado y la interpretación de la misma -->
<!-- A continuación mostramos el código original para la gestión de dos
capas “Answer1_Wrong” y “Answer1_Right” que proporcionan feedback cuando
se contesta una pregunta de manera correcta o incorrecta. La gestión
debe hacerse para tres estados excluyentes: show1-hide2, show2-hide1,
hide1-hide2 -->
```

```
<!-- in the file QUESTIONS -->

<html xmlns:IMS-LD="http://www.imsglobal.org/xsd/IMS-LD_v1p0"
xmlns="http://www.w3.org/1999/xhtml">

<div class="Answer1_Wrong">
      <p>This is not the right question. Make another choice</p>
</div>
<div class="Answer1_Right">
      <p>Congratulations! This is the right question. Look at the
picture</p>
      <img src="supporting-picture.jpg"></img>
</div>

<!-- in the file MANIFEST -->

<if>
      <is>
            <property-ref ref="Answer1"/>
            <property-value>A</property-value>
      </is>
</if>
<then>
      <show>
            <class class="Answer1_Right" />
      </show>
      <hide>
            <class class="Answer1_Wrong" />
      </hide>
</then>
      <else>
            <if>
                  <is-not>
                    <property-ref ref="Answer1"/>
                    <property-value>Default</property-value>
                  </is-not>
            </if>
            <then>
                  <show>
                        <class class="Answer1_Wrong" />
                  </show>
                  <hide>
                        <class class="Answer1_Right" />
                  </hide>
            </then>
                  <else>
                        <hide>
                              <class class="Answer1_Right" />
                        </hide>
                        <hide>
                              <class class="Answer1_Wrong" />
                        </hide>
                  </else>
      </else>
```

Ejemplo

```
<!-- El mismo ejemplo se simplificaría con la sintaxis siguiente,
asumiendo que las dos capas comienzan con la propiedad de visibilidad
desactivada -->

<!-- in the file QUESTIONS -->
```

```
<!-- … -->


<!-- in the file MANIFEST -->

<if>
      <is>
            <property-ref ref="Answer1"/>
            <property-value>A</property-value>
      </is>
</if>
<then>
      <show>
            <class class="Answer1_Right" />
      </show>
</then>
      <else>
            <if>
                  <is-not>
                    <property-ref ref="Answer1"/>
                    <property-value>Default</property-value>
                  </is-not>
            </if>
            <then>
                  <show>
                        <class class="Answer1_Wrong" />
                  </show>
            </then>
      </else>
```

ID	Elemento de análisis y/o crítica
Modelado	
[M.12]	No existe ningún soporte de iteración en ninguna de las estructuras básicas de la metáfora (actividad de aprendizaje, actividad de soporte, estructura de actividades, actos, obras)

Modificación y/o extensión propuestas	
[Ext.09]	Extender la sintaxis actual de cada elemento con un parámetro iteration que define una iteración numérica (tipo for) y/o una iteración por condición de entrada y repetición (tipo while)
Descripción	
Se añade a cada estructura citada un parámetro iteration que define el número de veces que se puede repetir como máximo (prop-ITER-MAX) o la condición de entrada y repetición de la ejecución de la actividad. Esta condición puede comparar valores específicos ad-hoc (property-value) o almacenados en variables (property-ref) Una repetición finaliza cuando se completa la estructura siguiendo cualquiera de los mecanismos contemplados actualmente (user-choice, time, property)	
Sintaxis	

```
<iteration>
  <is>
      <property-ref ref="Answer1"/>
      <property-value>VALUE</property-value>
  </is>
</iteration>

<!-- or -->

<iteration>
  <is>
```

```
      <property-ref ref="Answer1"/>
      <property-ref ref=”prop-condition”/>
  </is>
</iteration>


<!-- or -->

<iteration>
      <property-value>VALUE</property-value>
</iteration>

<!-- or -->

<iteration>
      <property-ref ref=”prop-ITER-MAX”/>
</iteration>

<!-- ... -->

<on-completion>
      <break/>
</on-completion>
```

Ejemplo

```
<!-- ... -->

<learning-activity isvisible="true" identifier="activity-1">
            <title>Activity to carry out</title>
            <activity-description>
                  <title>First part of the activity</title>
                  <item isvisible="true" identifierref="item-1"/>
            </activity-description>
            <iteration>
                  <property-value>3</property-value>
            </iteration>
</learning-activity>

<!-- ... -->

<play identifier="play-1" isvisible="true">
      <act identifier="act-1">
            <iteration>
              <is>
                  <property-ref ref="Answer1"/>
                  <property-value>A</property-value>
              </is>
            </iteration>
            <role-part identifier="rolepart-1">
                  <role-ref ref="student"/>
                  <learning-activity-ref ref="activity-1"/>
            </role-part>
      </act>
</play>
```

ID	Elemento de análisis y/o crítica
Modelado	
[M.13]	No existen puntos de entrada o sincronización dentro de la estructura del manifiesto

Modificación y/o extensión propuestas

[Ext.10]	Incorporar un elemento GOTO que permita la redirección directa del flujo de aprendizaje

Descripción

GOTO funciona como el clásico comando de programación a ser lanzado dentro de las estructuras on-completion de actividades de aprendizaje, estructuras de actividades, actos y obras o como consecuencia directa de una acción promovida por un servicio de notificación/trigger, una acción en una condición, etcetera

GOTO congela el estado actual de la ejecución antes del salto y lo continua en el lugar de continuacion. En caso de solapamiento de estados/variables, los valores de origen sustituyen a los valores de destino, complementándose en caso contrario (hay en destino variables que no hay en origen o viceversa)

Los puntos destino de entrada serán siempre cabeceras de actividad de aprendizaje, estructuras de actividades, acto u obra, ya que lo que se gestiona es el flujo de aprendizaje y no la circulación de contenido específico. No se permite por tanto saltar a un item concreto de una unidad de aprendizaje, por ejemplo. La cabecera de destino tiene que encontrase definida, no originando ninguna acción en caso contrario

El destino de GOTO puede estar indicado como valor explícito (property-value) o como contenido de propiedad tipo string (property-ref)

El salto afecta únicamente al usuario que lo ejecuta

No se pueden realizar saltos a elementos finalizados y cerrados (actos, actividades…)

Sintaxis

```
<!-- ... -->

<goto>
      <learning-activity-ref>
            <property-value>LA-1</property-value>
      </learning-activity-ref>

      <!-- or -->

      <learning-activity-ref>
            <property-ref ref="GOTO1"/>
      </learning-activity-ref>

      <!-- or -->

      <activity-structure-ref> ...

      <!-- or -->

      <act-ref> ...

      <!-- or -->

      <play-ref> ...

</goto>

<!-- ... -->
```

Ejemplo

```
<!-- ... -->

<locpers-property identifier="GOTO1">
      <datatype datatype="string"/>
      <initial-value>LA-1</initial-value>
</locpers-property>

<!-- ... -->
```

```
<if>
      <is>
            <property-ref ref="Answer1"/>
            <property-value>A</property-value>
      </is>
</if>
      <then>
            <goto>
                  <learning-activity-ref>
                        <property-ref ref="GOTO1"/>
                  </learning-activity-ref>
            </goto>
      </then>

<!-- ... -->
```

ID	Elemento de análisis y/o crítica
Modelado	
[M.14]	No se puede realizar la asignación concreta de una actividad específica a un usuario seleccionado

Modificación y/o extensión propuestas	
[Ext.11]	Incorporar un elemento ASSIGN-ACTIVITY-TO-USER que permite emparejar usuarios, grupos y roles con actividades de aprendizaje y estructuras de actividades

Descripción

El tutor (role staff) puede seleccionar un usuario/rol/grupo de una lista o mediante asignación directa en código y asignarle una determinada actividad de aprendizaje o una estructura de actividades, que estará inactiva para ese usuario/s en concreto. La tarea podrá ser activada y desactivada por el tutor en un momento concreto mediante otro comando SWITCH-ACTIVITY

Los usuarios, usuarios de grupos y usuarios de roles de la lista, incorporan la actividad o la estructura a su flujo de aprendizaje según la definición en el manifiesto

Es, por tanto, una tarea de administración y monitorización, referida al control de flujo de actividad

El proceso implica la generación automática de un role-part para cada usuario con una nueva actividado estructura asignadas, lo que supondría una modificación del manifiesto en runtime

Sintaxis

```
<!-- asigna la actividad LA-1 a los usuarios, roles y grupos de la lista
-->

<assign-activity-to-user>
      <learning-activity-ref ref="LA-1"/>
      <user-ref ref="USER-1"/>
      <user-ref ref="USER-2"/>
      <role-ref ref="ROLE-1"/>
      <group-ref ref="GROUP-1"/>
      <!-- ... -->
      <user-ref ref="USER-n"/>
</assign-activity-to-user>

<!-- or -->

<!-- asigna la estructura AS-1 a los usuarios, roles y grupos de la
lista -->
```

```
<assign-activity-to-user>
      <activity-structure-ref ref="AS-1"/>
      <user-ref ref="USER-1"/>
      <user-ref ref="USER-2"/>
      <role-ref ref="ROLE-1"/>
      <group-ref ref="GROUP-1"/>
      <!-- ... -->
      <user-ref ref="USER-n"/>
</assign-activity-to-user>

<!-- or -->

<!-- muestra todos los usuarios, roles, grupos, actividades de
aprendizaje y estructuras de actividades para su asignación -->

<assign-activity-to-user/>
```

Modificación y/o extensión propuestas	
[Ext.12]	Incorporar un elemento ASSIGN-USER-TO-ACTIVITY que permite emparejar usuarios, grupos y roles con actividades de aprendizaje y estructuras de actividades
Descripción	
El tutor (role staff) puede seleccionar una actividad o estructura de actividades de una lista o mediante asignación directa en código y asignarle un determinado usuario/rol/grupo. La tarea permanecerá inactiva para ese usuario/s por defecto, y podrá ser activada y desactivada por el tutor en un momento concreto mediante otro comando SWITCH-ACTIVITY Los usuarios, usuarios de grupos y usuarios de roles de la lista, incorporan la actividad o la estructura a su flujo de aprendizaje según la definición en el manifiesto Es, por tanto, una tarea de administración y monitorización, referida al control de flujo de actividad El proceso implica la generación automática de un role-part para cada usuario con una nueva actividado estructura asignadas, lo que supondría una modificación del manifiesto en runtime	
Sintaxis	

```
<!-- asigna el usuario USER-1 las actividades y estructuras de la lista
-->
<assign-user-to-activity>
      <user-ref ref="USER-1"/>
      <learning-activity-ref ref="LA-1"/>
      <activity-structure-ref ref="AS-1"/>
      <!-- ... -->
      <learning-activity-ref ref="LA-n"/>
</assign-user-to-activity>

<!-- or -->

<!-- asigna a todos los usuarios del rol ROLE-1 las actividades y
estructuras de la lista  -->

<assign-user-to-activity>
      <role-ref ref="ROLE-1"/>
      <learning-activity-ref ref="LA-1"/>
      <activity-structure-ref ref="AS-1"/>
      <!-- ... -->
      <learning-activity-ref ref="LA-n"/>
</assign-user-to-activity>

<!-- or -->
```

```
<!-- asigna a todos los usuarios del grupo GROUP-1 las actividades y
estructuras de la lista  -->

<assign-user-to-activity>
      <group-ref ref="GROUP-1"/>
      <learning-activity-ref ref="LA-1"/>
      <activity-structure-ref ref="AS-1"/>
      <!-- ... -->
      <learning-activity-ref ref="LA-n"/>
</assign-user-to-activity>

<!-- muestra todos los usuarios, roles, grupos, actividades de
aprendizaje y estructuras de actividades para su asignación -->

<assign-user-to-activity/>
```

Modificación y/o extensión propuestas	
[Ext.13]	Incorporar un elemento SWITCH-ACTIVITY que permite activar y desactivar actividades y estructuras de actividades

Descripción

El tutor (role staff) puede seleccionar una actividad o estructura de actividades de una lista o mediante asignación directa en código y activarla o desactivarla
Estas actividades y estructuras de actividades han sido previamente asignadas a usuarios, roles y grupos por ASSIGN-ACTIVITY-TO-USER y por ASSIGN-USER-TO-ACTIVITY

Sintaxis

```
<switch-activity>
      <on-off>
            <!-- true, false -->
            <property-value>true</property-value>
      </on-off>
      <learning-activity-ref>
            <property-value>LA-1</property-value>
      </learning-activity-ref>
      <activity-structure-ref>
            <property-ref ref="STRUCTURE-TO-SWITCH-1"/>
      </activity-structure-ref>
      <!-- ... -->
      <learning-activity-ref>
            <property-value>LA-n</property-value>
      </learning-activity-ref>
</switch-activity>

<!-- or -->

<switch-activity>
      <on-off>
            <property-ref ref="STATE"/>
      </on-off>
      <learning-activity-ref>
            <property-ref ref="ACTIVITY-TO-SWITCH-1"/>
      </learning-activity-ref>
      <activity-structure-ref>
            <property-value>AS-1</property-value>
      </activity-structure-ref>
      <!-- ... -->
      <learning-activity-ref>
            <property-ref ref="ACTIVITY-TO-SWITCH-n"/>
      </learning-activity-ref>
</switch-activity>
```

```
<!-- or -->

<!-- muestra una lista de actividades para activar-desactivar -->

<switch-activity/>
```

ID	Elemento de análisis y/o crítica
Modelado	
[M.15]	No existe la creación de grupos de manera selectiva dentro de la instancia

Modificación y/o extensión propuestas	
[Ext.14]	Incorporación de un elemento CREATE-GROUP que permite agrupar usuarios dentro de un mismo rol
Descripción	

El comando permite generar un grupo *ad hoc* con usuarios del mismo o distinto rol a seleccionar de una lista desplegable o mediante codificación. Este grupo puede luego ser utilizado en diversas acciones, como monitorización o asignación y activación de actividades
La lista de usuarios puede estar almacenada en una variable de tipo string
El nombre del grupo puede ser asignado directamente en código o ser el contenido de una variable

Sintaxis

```
<!-- muestra todos los usuarios, divididos por roles, y permite
asignarlos al grupo NAME-GROUP -->

<create-group>
      <group-name>
            <property-value>VALUE</property-value>
            <!-- or -->
            <property-ref ref="NAME-GROUP"/>
      </group-name>
</create-group>

<!-- asigna los usuarios indicados en la variable tipo string LIST-USERS
al grupo NAME-GROUP -->

<create-group>
      <group-name>
            <property-value>VALUE</property-value>
            <!-- or -->
            <property-ref ref="NAME-GROUP"/>
      </group-name>
      <user-ref ref="LIST-USERS"/>
</create-group>
```

ID	Elemento de análisis y/o crítica
Modelado	
[M.16]	No existe una estructura de trabajo colaborativo que no haya sido expresamente modelada con los recursos básicos existentes
[M.18]	No existe ningún sistema de comunicación entre usuarios/roles más allá de la monitorización uno a uno mediante la visualización de propiedades ajenas

Modificación y/o extensión propuestas	
[Ext.XX]	Ninguna
Descripción	

IMS-LD está diseñado para modelar metodologías pedagógicas y concreción de

planes de estudio, y no para desarrollar herramientas colaborativas o de comunicación que ya existen externamente. A IMS-LD cabe pedir la integración de dichas herramientas de manera ágil, sencilla y eficaz en el learning design. Referirse a [A.01] y [A.03], sobre el servicio de comunicación

ID	Elemento de análisis y/o crítica
Modelado	
[M.17]	El servicio de monitorización no contempla la monitorización grupal

Modificación y/o extensión propuestas	
[Ext.15]	Extensión del servicio de monitorización para seguimiento de roles y grupos

Descripción

El usuario de cualquier rol asignado para ello puede monitorizar el seguimiento de él mismo, otros usuarios, grupos y roles
Para grupos y roles, se pueden mostrar los valores únicos de cada usuario, o los valores comparados de todos los usuarios para la misma propiedad
También para grupos y roles, se pueden mostrar los valores de una propiedad o de todas las propiedades relacionadas
Ver extensión [Ext.07] sobre utilización de propiedades
Se modifica colateralmente la actividad de soporte relacionada con el entorno donde se ejecuta el servicio de monitorización, suprimiendo la referencia a un determinado role

Sintaxis

```
<!-- in the MANIFEST  -->

<service identifier="S-1">
      <monitor>
            <role-ref>
                  <property-ref ref="ROLE-1"/>
            </role-ref>
            <title>TITLE</title>
            <item identifierref="I-1"/>
      </monitor>
</service>

<service identifier="S-1">
      <monitor>
            <group-ref>
                  <property-value>VALUE</property-value>
            </group-ref>
            <title>TITLE</title>
            <item identifierref="I-1"/>
      </monitor>
</service>

<!-- ...  -->

<!-- in the XHTML external file -->

<!--visualiza la variable PROPERTY-1 del usuario único suported-user -->

      <view ref="PROPERTY-1" property-of="supported-user" />

<!--visualiza el contenido de la propiedad PROPERTY-1  para todos los
usuarios del grupo soportado -->

      <view ref="PROPERTY-1" property-of="supported-group" />
```

```
<!--visualiza el contenido de la propiedad PROPERTY-1  para todos los
usuarios del rol soportado -->

      <view ref="PROPERTY-1" property-of="supported-role" />

<!--visualiza todas las propiedades asociadas al usuario único
supported-user -->

      <view property-of="supported-user" />

<!--visualiza todas las propiedades asociadas a todos los usuarios del
grupo soportado -->

      <view property-of="supported-group" />

<!--visualiza todas las propiedades asociadas a todos los usuarios del
rol soportado -->

      <view property-of="supported-role" />
```

Ejemplo

```
<!-- in the MANIFEST -->

<!-- ... -->

<support-activity identifier="SA-first-step" isvisible="true">
      <title>Checking the submission of assesments</title>
      <role-ref ref="Learner"/>
      <environment-ref ref="E-overview-remarks-group"/>
      <activity-description>
            <item identifier="I-sa-first-step" identifierref="R-set-
activity2-complete">
                  <title>Close the submission</title>
            </item>
      </activity-description>
</support-activity>

<!-- ... -->

<environment identifier="E-overview-remarks-role">
      <title>Checking the submission</title>
      <service identifier="S-overview-submission">
            <monitor>
                  <group-ref ref="classroom-monday"/>
                  <title>What submissions?</title>
                  <item identifierref="R-submission-overview"/>
            </monitor>
      </service>
</environment>

<!-- ... -->

<role-part identifier="RP-Tutor-1">
      <title>Checking the submission process</title>
      <role-ref ref="Tutor"/>
      <support-activity-ref ref="SA-first-step"/>
</role-part>

<!-- ... -->

<resource identifier="R-set-activity2-complete" type="imsldcontent"
```

```
href="set-activity2-complete.xml">
      <file href="set-activity2-complete.xml"/>
</resource>

<!-- in the XHTML external file "set-activity2-complete.xml"-->

Esta es una propiedad de cada usuario:
      <view ref="PROPERTY-1" property-of="supported-user" />

Esta es una propiedad de todos los usuarios del grupo "classroom-
monday":
      <view ref="PROPERTY-1" property-of="supported-group" />

Estas son todas las propiedades de cada usuario:
      <view property-of="supported-user" />

Estas son todas las propiedades de todos los usuarios del grupo
"classroom-monday":
      <view property-of="supported-group" />
```

ID	Elemento de análisis y/o crítica
Arquitectura	
[A.01]	Para lograr la interoperabilidad con otras especificaciones y módulos externos a IMS-LD es necesario un sistema o capa de comunicación que permita el intercambio efectivo de información, como variables y valores de variables y estado, lo que permitirá interactuar desde IMS-LD en el módulo externo y viceversa
[A.03]	Se hace preciso el desarrollo de una capa de comunicación o dispatcher para lograr la integración de módulos externos con IMS-LD

Modificación y/o extensión propuestas	
[Ext.XX]	Ninguna
Descripción	
Se buscan dos cosas: La interoperabilidad y la integración de módulos externos. Para ambas es necesario desarrollar una capa de comunicación o dispatcher. Existe un primer acercamiento a través de la capa CCSI (CopperCore Service Integration) [234]. CCSI permite la comunicación de IMS-LD con el exterior. Esta capa ha sido utilizada como ejemplo para la integración de juegos en Unidades de Aprendizaje adaptativas, tal y como ya se ha descrito en la Sección 3.11, mediante trabajo conjunto con la Universidad Complutense de Madrid [161]. En esta sección se describen los problemas y las soluciones proporcionadas, así como también se aportan referencias bibliográficas relacionadas. No es el objetivo de esta tesis adentrarse en la crítica y extensión de CCSI sino de la especificación IMS-LD en sí. CCSI supone una extensión de IMS-LD utilizando el motor CopperCore y no forma parte de la especificación. Del mismo modo, cualquier otro sistema que no utilice CopperCore para validar, publicar, interpretar y ejecutar Unidades de Aprendizaje escritas en IMS-LD no puede hacer uso de CCSI.	

ID	Elemento de análisis y/o crítica
Arquitectura	
[A.02]	IMS-LD no permite la grabación ni la recuperación de datos en y desde ficheros externos de ningún formato. Del mismo modo, las conexiones con bases de datos externas o con módulos programados en otros lenguajes no están contempladas dentro de la especificación

Modificación y/o extensión propuestas

[Ext.16] [Ext.17] [Ext.18]	Incorporación de un elemento EXPORT y de un elemento IMPORT que gestionarán un fichero de tipo parametrizadao (tipo TXT, por ejemplo) declarado en una propiedad de un tipo nuevo FILE-IO

Descripción

El fichero se declarará en una propiedad de tipo FILE-IO. La declaración hará referencia a un fichero existente de tipo CSV almacenado local o remotamente
El camino del fichero se indicará de manera absoluta, vía local C:\ o remota HTTP://
En caso de comunicación remota el envío-recepción de variables puede funcionar por cabecera como método GET-POST, habilitando la comunicación de variables entre la Unidad de Aprendizaje y el exterior
La comprobación de que el fichero existe se realizará en el momento de validación de la UoL, proporcionando un mensaje de error adecuado en caso de anomalía
La estructura se compone del nombre de la variable que almacena el fichero y una lista de variables a grabar-recuperar. En caso de ausencia de esta lista, se graban o se recuperan todas las variables que afecten al usuario. De facto, esto representa un backup del estado actual del usuario
La sintaxis del fichero será definida por pares VARIABLE:VALOR

Sintaxis

```
<!-- ... -->

<import identifier="var-import">
      <property-ref ref="var-file"/>
</import>

<!-- ... -->

<import identifier="var-import">
      <property-ref ref="var-file"/>
      <property-ref ref="var1"/>
      <!-- ... -->
      <property-ref ref="varN"/>
</import>

<!-- ... -->

<export identifier="var-export">
      <property-ref ref="var-file"/>
</export>

<!-- ... -->

<export identifier="var-export">
      <property-ref ref="var-file"/>
      <property-ref ref="var1"/>
      <!-- ... -->
      <property-ref ref="varN"/>
</export>
```

Ejemplo

```
<!-- ... -->

<locpers-property identifier="info-in-file-local">
      <datatype datatype="file-io"/>
      <initial-value>C:\file-with-info.txt</initial-value>
</locpers-property>

<locpers-property identifier="info-in-file-remote">
      <datatype datatype="file-io"/>
      <initial-value>http://www.ou.nl/file-with-info.txt</initial-value>
</locpers-property>
```

```
<locpers-property identifier="name">
      <datatype datatype="string"/>
</locpers-property>

<locpers-property identifier="age">
      <datatype datatype="integer"/>
</locpers-property>

<global-property identifier="threshold">
      <datatype datatype="real"/>
</global-property>

<!-- ... -->

<export identifier="save-local-all">
      <property-ref ref="info-in-file-local"/>
</export>

<export identifier="save-local-some">
      <property-ref ref="info-in-file-local"/>
      <property-ref ref="name"/>
      <property-ref ref="age"/>
      <property-ref ref="threshold"/>
</export>

<!-- ... -->

<import identifier="load-remote-all">
      <property-ref ref="info-in-file-remote"/>
</import>

<import identifier="load-remote-some">
      <property-ref ref="info-in-file-remote"/>
      <property-ref ref="name"/>
      <property-ref ref="age"/>
      <property-ref ref="threshold"/>
</import>

<!-- ... -->

<!-- format of the file

name:ambjorn-naeve
age:49
threshold:10.0

end format of the file -->
```

Modificación y/o extensión propuestas	
[Ext.19] [Ext.20] [Ext.21]	Incorporación de un elemento FROM-DB y de un elemento TO-DB que permiten recuperar y almacenar información de una base de datos en formato MySQL (inicialmente). La conexión de define en una propiedad de un tipo nuevo DATABASE
Descripción	
En la declaración de propiedades se define una propiedad de un tipo nuevo DATABASE con la configuración de una base de datos de tipo MySQL (o cualquier otro estándar) La lectura-grabación de información podría relacionarse con diversos servicios: Gestión de usuarios y roles, volcado-recuperación de contenidos de variables En FROM-DB la query se define en una variable de tipo string que forma parte de la	

sintaxis de la estructura. En este caso, la estructura se compone del nombre de la variable con la base de datos, el nombre de la variable con la consulta y la lista de variables que almacenarán la recuperación Funcionamiento y sintaxis similares a lo descrito en [Ext.16], [Ext.17] y [Ext.18], a ser concretadas según el tipo de base de datos a utilizar con información para grabar-recuperar, el tipo de base de datos del motor de ejecución (por ejemplo, Coppercore) y su gestión de usuarios
Ejemplo

```
<!-- ... -->

<locpers-property identifier="info">
      <datatype datatype="database"/>
      <initial-value>C:\database.db</initial-value>
      <format>mysql</format>
</locpers-property>

<locpers-property identifier="info">
      <datatype datatype="database"/>
      <initial-value>http://www.ou.nl/database.db</initial-value>
</locpers-property>

<locpers-property identifier="query">
      <datatype datatype="string"/>
      <initial-value>retrieval-query</initial-value>
</locpers-property>

<locpers-property identifier="name">
      <datatype datatype="string"/>
</locpers-property>

<locpers-property identifier="age">
      <datatype datatype="integer"/>
</locpers-property>

<global-property identifier="threshold">
      <datatype datatype="real"/>
</global-property>

<!-- ... -->

<to-db identifier="save-local-all">
      <property-ref ref="info"/>
</to-db>

<to-db identifier="save-local-some">
      <property-ref ref="info"/>
      <property-ref ref="name"/>
      <property-ref ref="age"/>
      <property-ref ref="threshold"/>
</export>

<!-- ... -->

<from-db identifier="load-remote-all">
      <property-ref ref="info"/>
</from-db>

<from-db identifier="load-remote-some">
      <property-ref ref="info"/>
      <property-ref ref="query"/>
      <property-ref ref="name"/>
```

```
      <property-ref ref="age"/>
      <property-ref ref="threshold"/>
</from-db>

<!-- ... -->
```

ID	Elemento de análisis y/o crítica
Arquitectura	
[A.04]	No hay posibilidad de modificar el esqueleto, el método, la definición de roles o cualquier otro elemento estructural en la fase de ejecución o run-time

Modificación y/o extensión propuestas	
[Ext.22] [Ext.23] [Ext.24] [Ext.25]	Se incorporan dos pares de *global elements* a) *view-IMS-LD* y *set-IMS-LD*, b) *view-resources* y *set-resources*, permitiendo visualizar y modificar la estructura del learning design y los recursos asignados en tiempo de ejecución
Descripción	
Es un servicio incorporado de administración, orientado para el rol de tipo Staff Relacionado con la extensión [Ext.07] sobre *global elements* Los elementos view-IMS-LD y set-IMS-LD permiten visualizar y modificar la estructura del learning design en tiempo de ejecución. De esta manera, al invocarlos dentro de un fichero externo tipo IMS-LD se mostrará la relación entre plays, acts, activity structures y activities, así como la asignación de role-parts con determinados roles y activities o activity structures asociados. Con set-IMS-LD, además podrá ser modificado. Para ello, el sistema tendrá en cuenta el estado actual de los usuarios conectados y de las variables de ejecución, bloqueando la modificación en caso de que alguna actividad se encuentre incompleta o sea incompatible Puede proporcionar la posibilidad de bloqueo de los usuarios y de su ejecución para realizar reformas necesarias que afecten globalmente, aunque ello suponga la pérdida de algunos estados de variables y usuarios. Es el caso de la modificación en tiempo real de errores de diseño Los elementos view-resources y set-resources permiten a un usuario de tipo rol Staff visualizar un listado que indica que recursos se asocian a qué ficheros y de qué tipo, pudiendo modificar los ficheros y el tipo de los mismos. Es decir, no se cambian los identificadores del recurso ni sus punteros, sino el fichero enlazado Ambos pares de global elements implican un cambio en la edición y ejecución de las Unidades de Aprendizaje tal y como se realizan hasta hoy. De esta manera, la UoL sería interpretada en tiempo real y no compilada, por lo que las diversas fases de diseño, validación, publicación y ejecución deberían verse modificadas y relacionadas de manera dinámica	
Sintaxis	

```
<view-IMS-LD/>

<set-IMS-LD/>

<view-resources/>

<set-resources/>
```

7. Evaluación de la tesis

7.1 . Método y enfoque

La evaluación de la tesis doctoral y de la solución desarrollada abarca dos partes:

a) Parte consultiva con expertos en diseño EML. Se contacta con un selecto grupo de expertos en lenguajes de modelado y especificaciones eLearning. Se recopila su opinión sobre la tesis, la solución de modelado, la solución de arquitectura, las unidades de aprendizaje y los casos de estudio desarrollados en la parte empírica. Fundamentalmente, si estas extensiones con los objetivos propuestos y si las soluciones mejoran la expresividad de IMS-LD en adaptación e interoperabilidad

b) Parte consultiva con diseñadores de aprendizaje y tutores online en activo. Se contacta un amplio y variado grupo de usuarios de sistemas de aprendizaje con conocimiento de IMS-LD y se recopila su opinión y crítica sobre los escenarios de aprendizaje modelados en la parte empírica con las extensiones y modificaciones desarrolladas en la tesis. Al igual que con los expertos en diseño EML, se pretende valorar si se cumplen los objetivos y si las soluciones de modificación y extensión mejoran la adaptación e interoperabilidad que IMS-LD puede modelar; esta vez, desde el punto de vista práctico de los usuarios cotidianos de sistemas y procesos online en entornos reales de aprendizaje

En vez de realizar un prototipo que ejecutara las modificaciones y extensiones propuestas, hemos decidido realizar una evaluación basada en el contraste de la investigación con expertos en la materia desde distintos puntos de vista. Los motivos son varios:

a) Supondría centrar los esfuerzos en la modificación efectiva del esquema de IMS-LD, sin haber realizado al menos un contraste con el organismo de estandarización asociado. Es decir, podría ser un trabajo teórico sin sustento de viabilidad. Este contraste debe necesariamente enmarcarse en un grupo o proyecto de desarrollo ajeno a los grupos de trabajo donde se ha desarrollado la tesis

b) Del mismo modo, se aportaría una solución práctica de programación, ya que habría que construir un engine y un player capaz de ejecutar la nueva versión de la especificación resultado de la tesis. Este esfuerzo se sitúa en otra capa de trabajo de más bajo nivel, vinculada con nuestra investigación pero fuera de los objetivos

c) A su vez, esto supondría un enfoque completamente distinto de lo que se pretende aquí, ya que el esfuerzo se centra en la definición conceptual y en la propuesta de extensiones y modificaciones genéricas, basadas en problemas específicos; no así en proporcionar una respuesta *ad hoc* no estandarizada. Es decir, si la propuesta es desarrollada individualmente, se vuelve a caer en una de las deficiencias que se estudian, como es la falta de interoperabilidad e integración de las Unidades de Aprendizaje, ya que proporcionaríamos una solución local

d) El esfuerzo de programación, ejecución y depuración se incluye en la fase de análisis con el modelado y prueba de diversas Unidades de Aprendizaje que abarcan todos los niveles de IMS-LD

e) En el estado actual, la realización de un prototipo aportaría poco o nada a la demostración de la efectividad y adecuación de las soluciones propuestas, ya que estas deben pasar por diversas etapas (organismo regulador, grupos de desarrollo, test de interoperabilidad…) antes de que puedan ver la luz

Sin embardo, la consulta con un grupo selecto de profesionales que utilizan y modelan Unidades de Aprendizaje con IMS-LD supone un contraste efectivo con la realidad actual del objeto de estudio, obviando los largos procesos de modificación de versiones de especificaciones. Este proceso se incluirá como parte del trabajo futuro a realizar y continuación de esta tesis.

Con el objetivo de obtener un *feedback* útil y significativo, hemos procedido a enviar 100 cuestionarios a otros tantos profesionales de los perfiles ya descritos dentro de Europa [235-237]. Hemos recibido 64 cuestionarios rellenados y válidos que aportan los resultados que se muestran en la siguiente sección. Además de una ficha tipo para encuadrar el perfil y áreas de interés e investigación del encuestado, existen catorce grupos de preguntas, haciendo un total de 142 subapartados.

7.2 . **Resultados e interpretación**

Los distintos apartados, sus resultados más destacados y la lectura asociada son los siguientes. Los resultados completos se adjuntan en un apéndice final *(Nota: Las cifras se indican en porcentaje sobre una muestra de 64 individuos; las opciones de los subapartados no son excluyentes; error ±0,1)*

Sample description: research field, profile

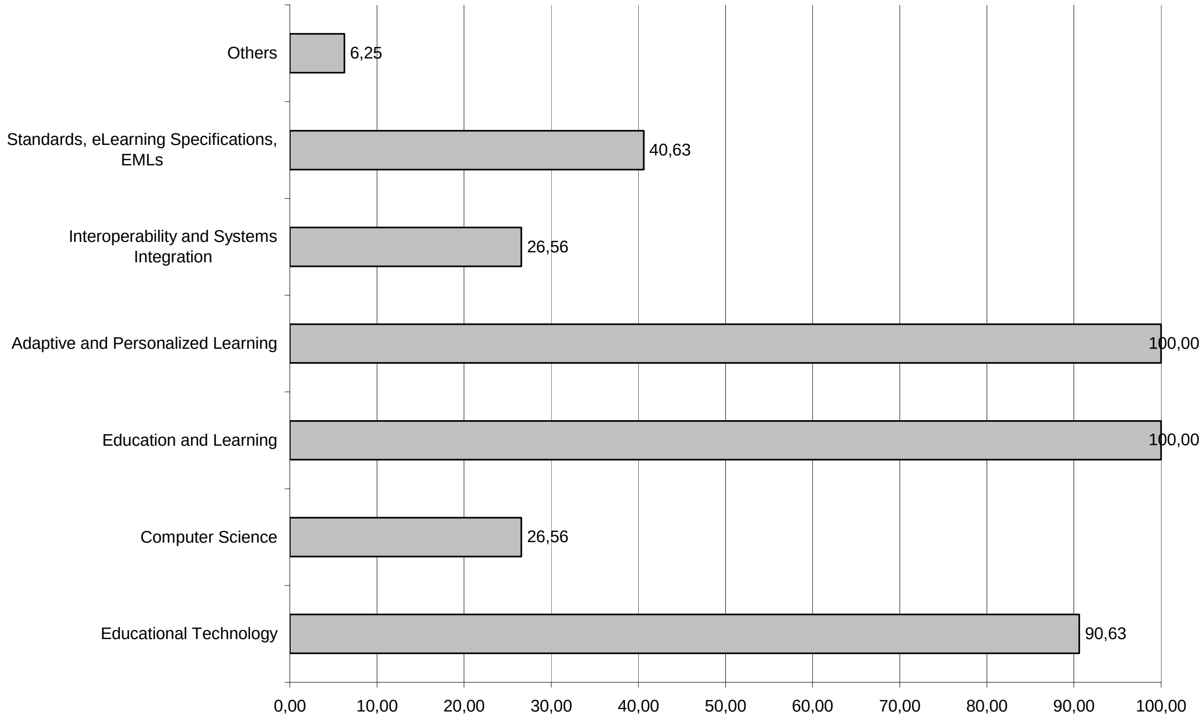


Figure 51. Evaluation: research fields

Lo primero que deducimos de esta ilustración que muestra los campos de investigación (Figure 51) es que entre el 90% y el 100% de los encuestados trabaja en tecnología educativa, educación o aprendizaje personalizado. Es decir, completamente centrado en el tema objeto de la tesis. Lo segundo es que más del 40% trabaja con estándares, lenguajes de modelado y especificaciones eLearning. También que más del 26% se centra en algo tan específico como es la interoperabilidad e integración de sistemas. Recordemos que la mitad de la muestra estaba orientada a público específicamente técnico, concretamente expertos en diseño EML. Esto abarca informáticos en sentido amplio, pero también profesionales de formación mixta que trabajan en tecnología educativa, a caballo entre la técnica y la educación. De la lectura anterior se desprende la perfecta adecuación de la muestra consultada con el objeto de la tesis.

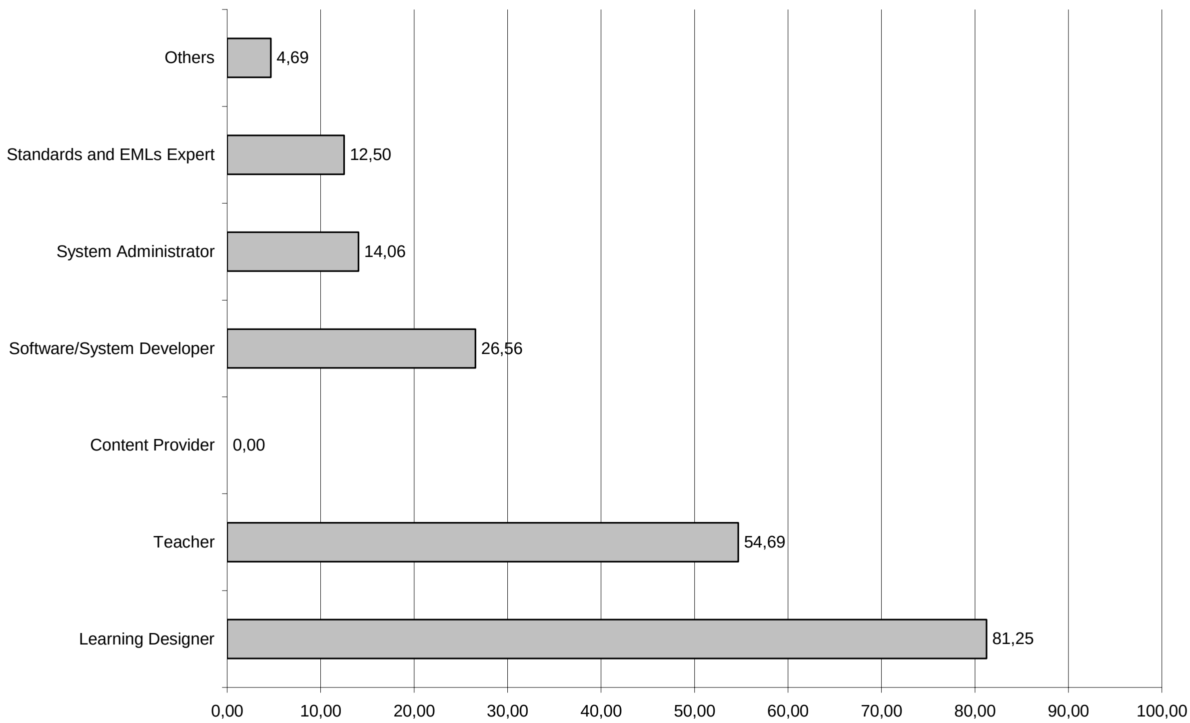
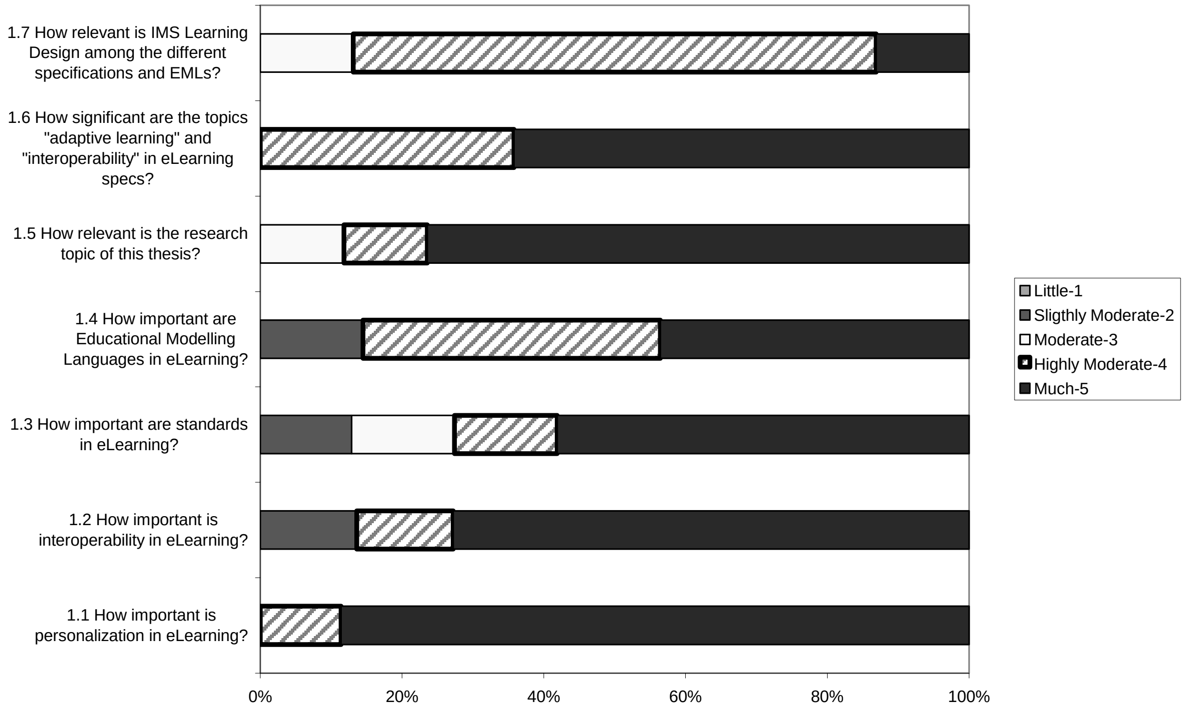


Figure 52. Evaluation: profiles

Esta segunda ilustración (Figure 52) constata que más del 80% de los encuestados son diseñadores de aprendizaje y más del 50% profesores. Esto es, los perfiles de primer orden relacionados con el objeto de la tesis. Del mismo modo, comprobamos como la suma de perfiles técnicos (que fruto del estudio de los cuestionarios se demuestran como opciones excluyentes en este caso) es de más del 50%. Es decir, expertos en especificaciones, lenguajes de modelado, administradores de sistema y desarrolladores. Nuevamente, un *target* muy alto relacionado directamente con el objeto de la tesis. Por el contrario, no observamos ningún proveedor de contenido. Aunque esta opción fue introducida en el cuestionario, el resultado obtenido concuerda con el esperado, dado que el público objetivo tiende a considerar que su trabajo se sitúa en nivel más conceptual y diferente al de creador de contenidos. Simplemente, constatamos que nuestra teoría es acertada para esta muestra.

1. About the topic and the context

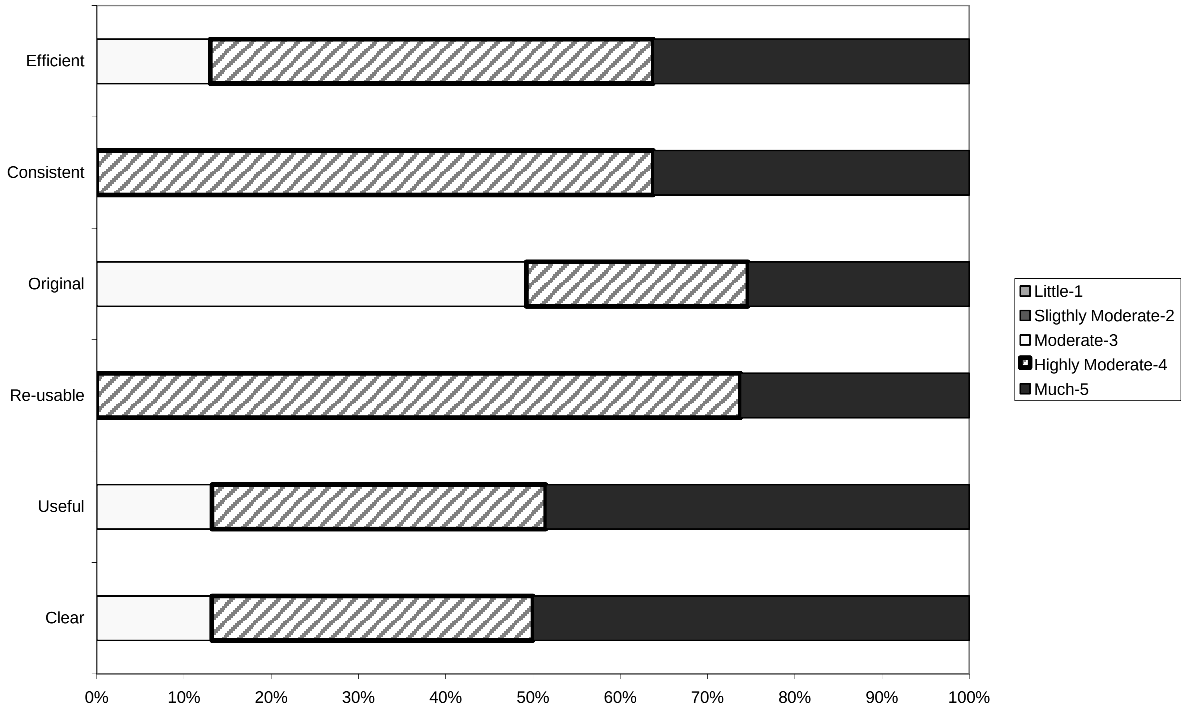


	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
1.1 How important is personalisation in eLearning?	0,0	0,0	0,0	**11,4**	**88,6**	4,89	0,32
1.2 How important is interoperability in eLearning?	0,0	13,6	0,0	**13,6**	**72,7**	4,45	1,04
1.3 How important are standards in eLearning?	0,0	12,9	14,5	**14,5**	**58,1**	4,18	1,11
1.4 How important are Educational Modeling Languages in eLearning?	0,0	14,5	0,0	**41,9**	**43,5**	4,14	1
1.5 How relevant is the research topic of this thesis?	0,0	0,0	11,8	**11,8**	**76,5**	4,65	0,69
1.6 How significant are the topics "adaptive learning" and "interoperability" in eLearning specs?	0,0	0,0	0,0	**35,8**	**64,2**	4,64	0,48
1.7 How relevant is IMS Learning Design among the different specifications and EMLs?	0,0	0,0	13,1	**73,8**	**13,1**	4,00	0,51

Table 13. Adecuación del tema y del contexto de investigación

En esta primera pregunta del cuestionario sobre la adecuación del tema y del contexto de investigación en sí (Table 13) comprobamos que existen unas cifras altas (Highly moderate-4) y muy altas (Much-5) para todos los subapartados, pasando desde el 72% que considera que los estándares son importantes para el aprendizaje online, hasta el 100% que valora el aprendizaje personalizado con la importancia en este rango alto o muy alto. Del mismo modo, el 100% considera que aprendizaje adaptativo e interoperabilidad tienen una importancia alta o muy alta.

2. The methodology used in this thesis to research about the topic is

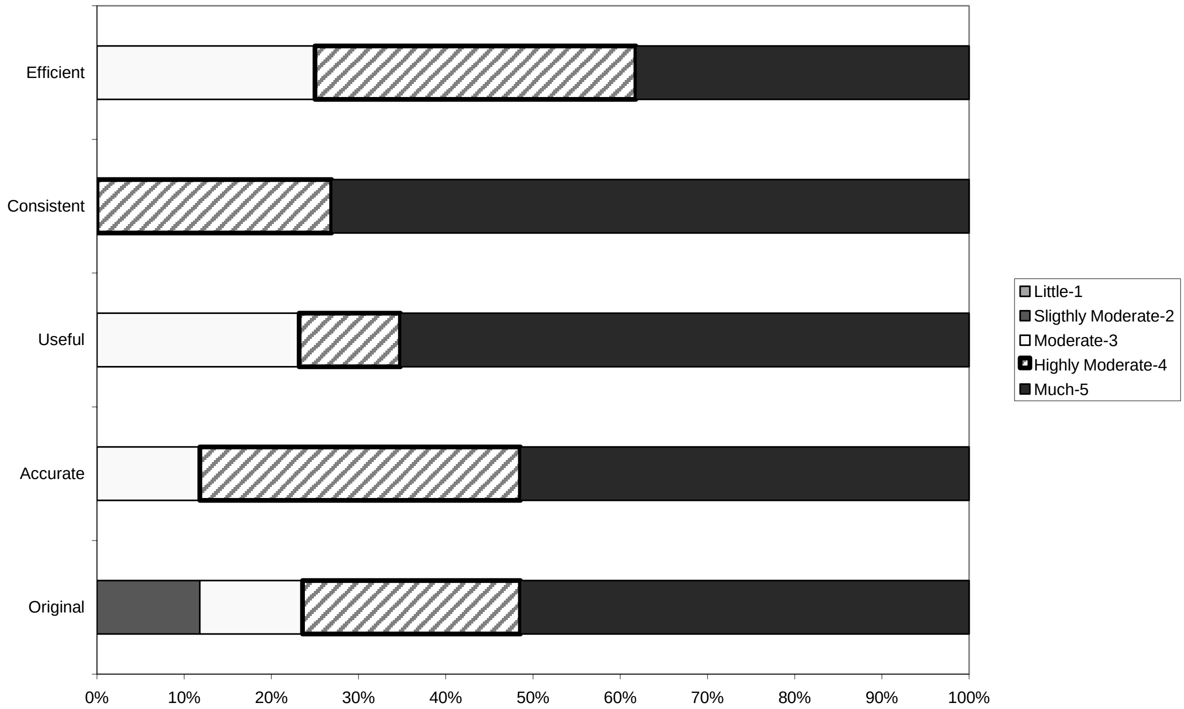


	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Clear	0,0	0,0	13,2	**36,8**	**50,0**	4,37	0,71
Useful	0,0	0,0	13,2	**38,2**	**48,5**	4,35	0,71
Re-usable	0,0	0,0	0,0	**73,8**	**26,2**	4,26	0,44
Original	0,0	0,0	**49,3**	25,4	25,4	3,77	0,83
Consistent	0,0	0,0	0,0	**63,8**	**36,2**	4,36	0,48
Efficient	0,0	0,0	13,0	**50,7**	**36,2**	4,23	0,67
					Avg	4,22	0,64

Table 14. Metodología utilizada en esta tesis

La metodología utilizada en la tesis (Table 14) ha sido considerada por más del 86% como clara, útil, reutilizable, consistente y eficiente, en grado alto o muy alto. La originalidad, sin embargo, se encuentra divididad entre los grados moderado, por un lado, y alto o muy alto, por otro.

3. The analysis of the current research context in the thesis is

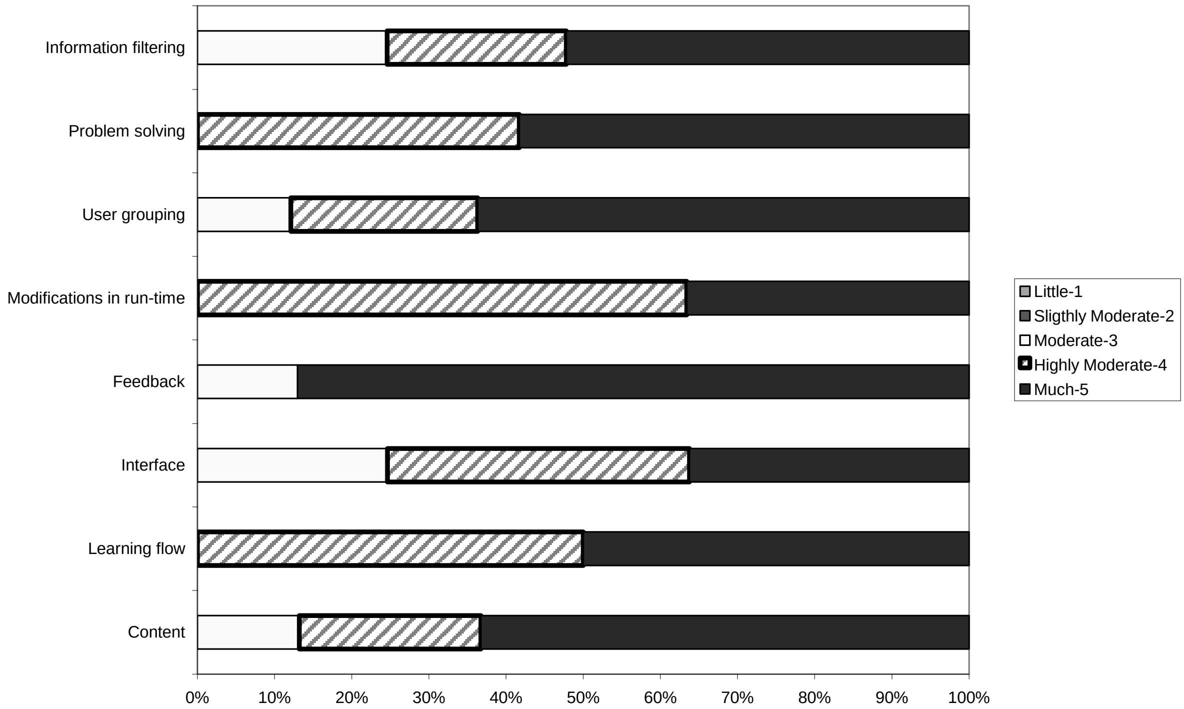


	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Original	0,0	**11,8**	11,8	**25,0**	**51,5**	4,17	1,04
Accurate	0,0	0,0	11,8	**36,8**	**51,5**	4,40	0,69
Useful	0,0	0,0	23,2	**11,6**	**65,2**	4,42	0,85
Consistent	0,0	0,0	0,0	**26,9**	**73,1**	4,73	0,45
Efficient	0,0	0,0	25,0	**36,8**	**38,2**	4,13	0,79
					Avg	4,37	0,76

Table 15. Análisis del contexto actual de investigación

El análisis del contexto actual de investigación (Table 15) ha sido considerado como alto o muy alto por más del 75% de los encuestados en cada subapartado. Destacar que el 11,8% lo ha considerado ligeramente bajo en originalidad.

4. How relevant are each of these types of adaptation in eLearning?

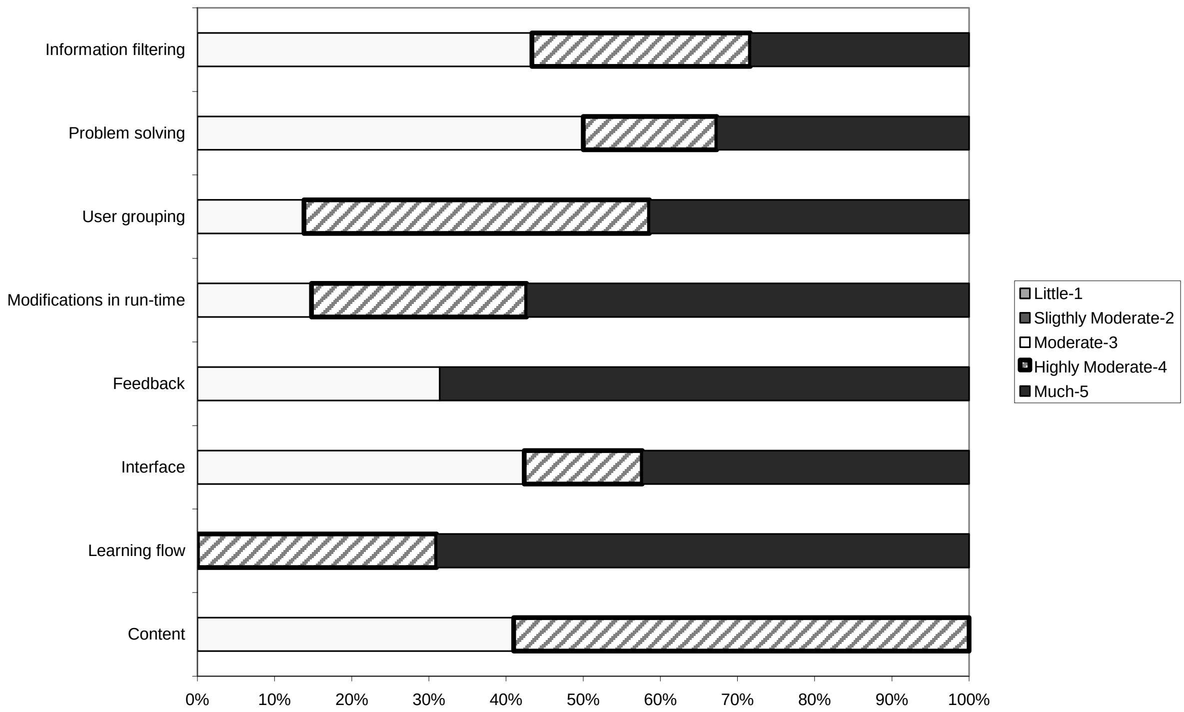


	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Content	0,0	0,0	13,2	**23,5**	**63,2**	4,50	0,72
Learning flow	0,0	0,0	0,0	**50,0**	**50,0**	4,50	0,50
Interface	0,0	0,0	24,6	**39,1**	**36,2**	4,11	0,77
Feedback	0,0	0,0	13,0	0,0	**87,0**	4,74	0,68
Modifications in run-time	0,0	0,0	0,0	**63,4**	**36,6**	4,37	0,48
User grouping	0,0	0,0	12,1	**24,2**	**63,6**	4,51	0,70
Problem solving	0,0	0,0	0,0	**41,7**	**58,3**	4,58	0,50
Information filtering	0,0	0,0	24,6	**23,2**	**52,2**	4,28	0,84
					Avg	4,45	0,65

Table 16. Relevancia de los tipos de adaptación relacionados con el aprendizaje

De los diversos tipos de adaptación considerados en la tesis (Table 16) se deduce que todos tienen una importancia alta o muy alta, destacando el 100% de *learning flow* y de *modifications in run-time*. Mientras que el learning flow es parte intrínseca de la especificación, las modificaciones en tiempo de ejecución son un reto para las herramientas que interpretan la especificación, pero no para ella misma. Del mismo modo, llama la atención la alta importancia que se otorga a *problema solving*, una vez más el 100%, que se clasifica dentro de nuestra investigación como un tipo depediente del *learning flow*. El *feedback* adaptativo obtiene un 87% de importancia muy alta. A él le dedicamos gran parte de la tesis, como parte del aprendizaje adaptativo y de la modificación del flujo de aprendizaje.

5. How relevant is the solution provided in this thesis and focused on ADAPTIVE LEARNING, related to these eight types of adaptation?



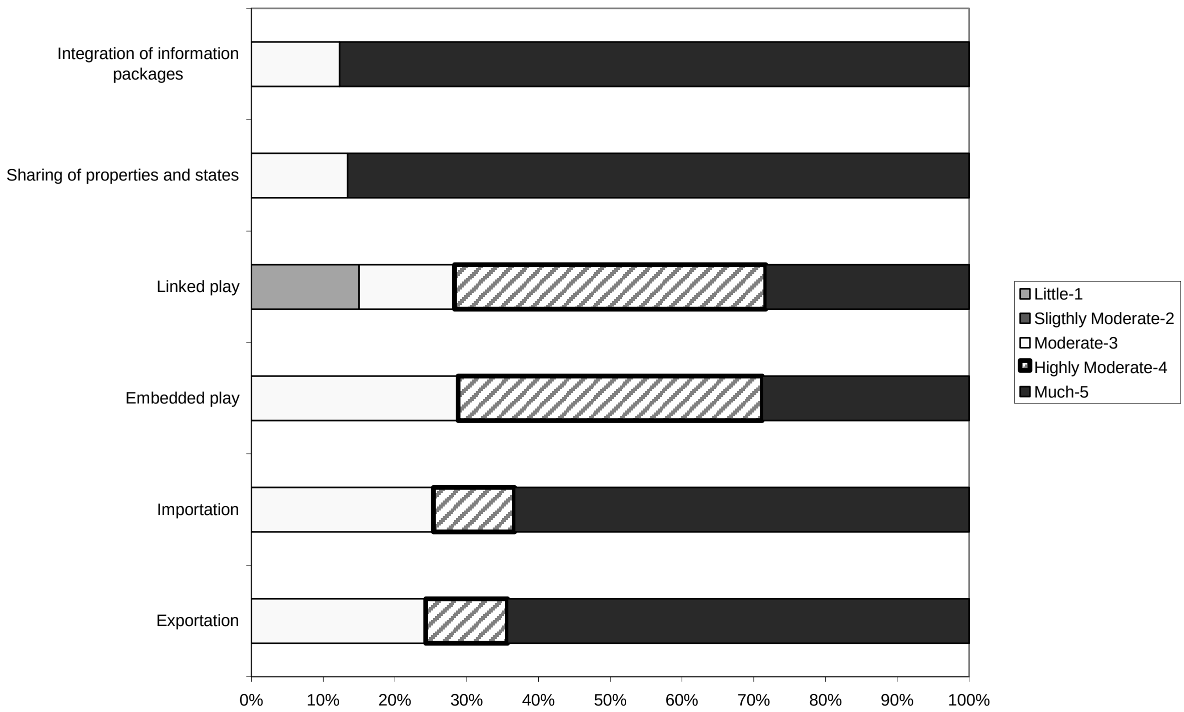
	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Content	0,0	0,0	**41,0**	**59,0**	0,0	3,59	0,49
Learning flow	0,0	0,0	0,0	**31,0**	**69,0**	4,69	0,46
Interface	0,0	0,0	**42,4**	15,3	**42,4**	4,00	0,93
Feedback	0,0	0,0	**31,4**	0,0	**68,6**	4,37	0,93
Modifications in run-time	0,0	0,0	14,8	**27,9**	**57,4**	4,43	0,74
User grouping	0,0	0,0	13,8	**44,8**	**41,4**	4,28	0,69
Problem solving	0,0	0,0	**50,0**	17,3	**32,7**	3,83	0,90
Information filtering	0,0	0,0	43,3	**28,3**	**28,3**	3,85	0,84
					Avg	4,13	0,75

Table 17. Relevancia de la solución aportada respecto del aprendizaje adaptativo

La solución aportada, en relación con cada tipo de adaptación (Table 17), muestra que los tres principales tipos (*Content*, *Learning Flow* e *Interface*) son evaluados al 100%

entre moderado y muy alto, destacando el 100% del contenido, como moderado o alto, el 100% del flujo de aprendizajem como alto o muy alto. También destacamos que ninguna de las aportaciones ha sido valorada con relevancia baja

6. How important are each of these ways of interoperability in eLearning?



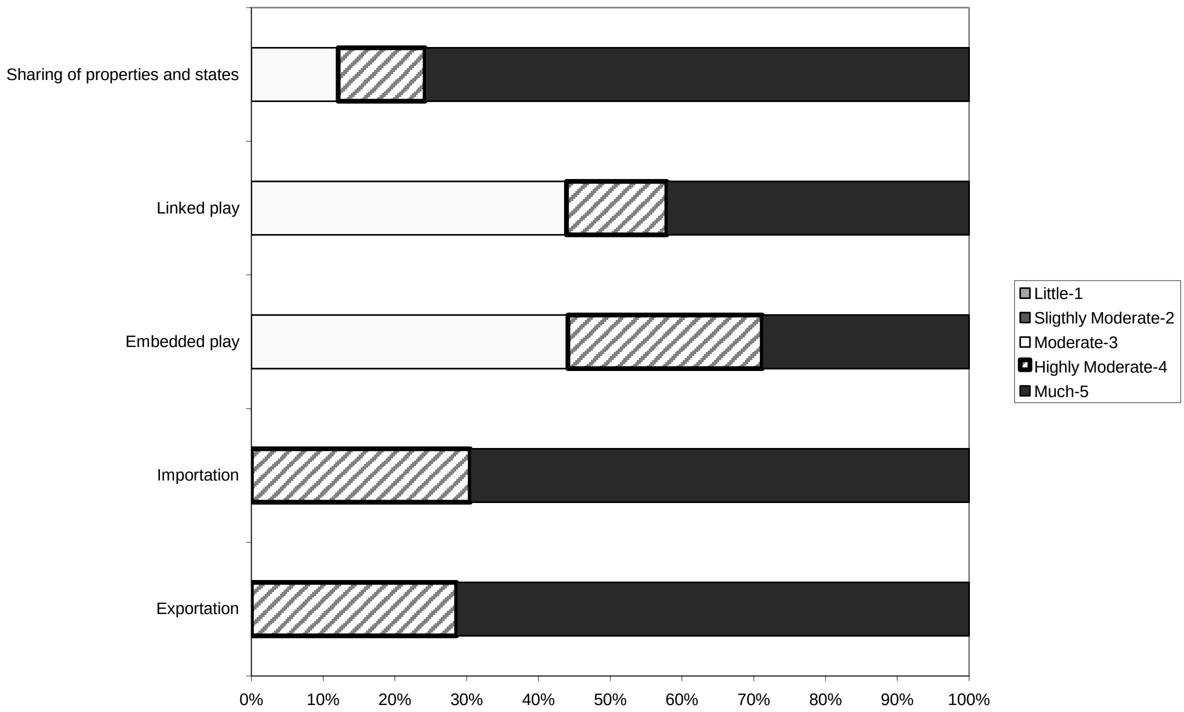
	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Exportation	0,0	0,0	24,3	11,4	**64,3**	4,40	0,86
Importation	0,0	0,0	25,4	11,3	**63,4**	4,38	0,87
Embedded play	0,0	0,0	28,8	**42,4**	**28,8**	4,00	0,76
Linked play	15,0	0,0	13,3	**43,3**	**28,3**	3,70	1,30
Sharing of properties and states	0,0	0,0	13,4	0,0	**86,6**	4,73	0,68
Integration of information packages	0,0	0,0	12,3	0,0	**87,7**	4,75	0,66
					Avg	4,33	0,86

Table 18. Importancia de los distintos tipos de interoperabilidad en aprendizaje online

La importancia de los distintos tipos de interoperabilidad utilizados en aprendizaje online (Table 18) se valora de distinta manera, probablemente coincidiendo con el tipo de uso que realiza cada usuario. Así, la compartición de propiedades y estados, y la integración de paquetes de información obtienen una importancia muy alta (por encima del 86%); la importación y la exportación una importancia muy alta (por encima del 63%); y la ejecución enlazada y embebida una importancia alta o muy alta (por encima del 71%).

Destaca por abajo la consideración de poca importancia para un 15% de los encuestados (sobre la muestra de 64 individuos, eso supone casi 10) de la ejecución enlazada. Tal y como describimos en la tesis, este tipo de ejecución soluciona de manera muy trivial la integración de unidades de aprendizaje, mientras que la compartición y la integración de paquetes destacan por lo contrario.

7. How relevant is the solution provided in this thesis and focused on INTEGRATION of Units of Learning, related to these ways?



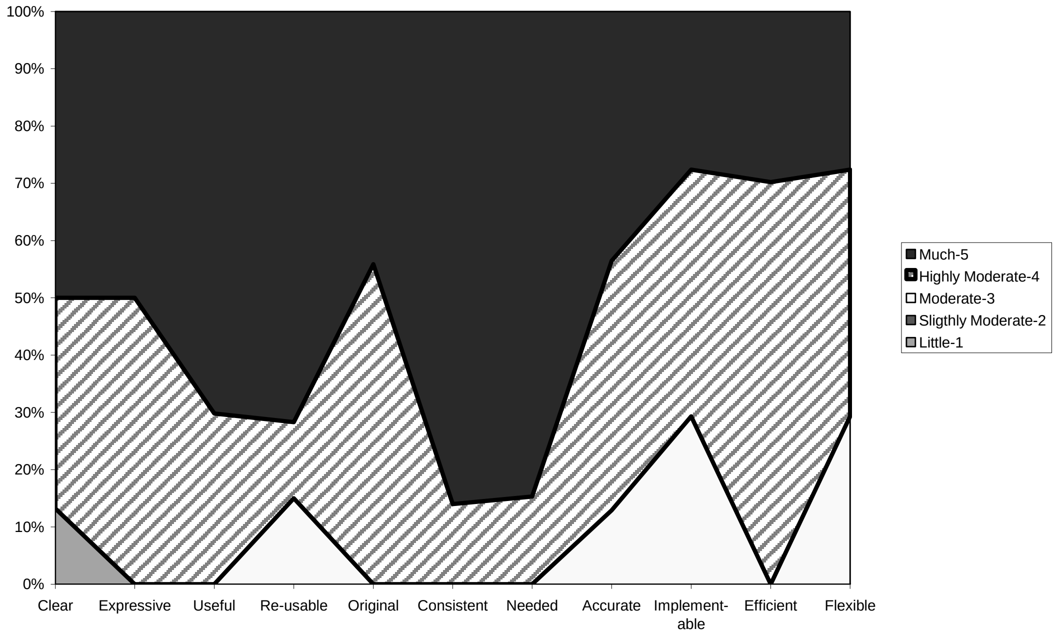
	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Exportation	0,0	0,0	0,0	**28,6**	**71,4**	4,71	0,45
Importation	0,0	0,0	0,0	**30,5**	**69,5**	4,70	0,46
Embedded play	0,0	0,0	**44,1**	27,1	**28,8**	3,85	0,84
Linked play	0,0	0,0	**43,9**	14,0	**42,1**	3,98	0,93
Sharing of properties and states	0,0	0,0	12,1	**12,1**	**75,8**	4,64	0,69
					Avg	4,38	0,67

Table 19. Relevancia de la solución centrada en la integración de Unidades de Aprendizaje

El 100% opina que la solución aportada sobre exportación e importación es satisfactoria en grado alto o muy alto (Table 19). Eso supone un 100% del objetivo de la tesis en este apartado. En menor medida, hemos tratado la ejecución, aunque nunca como objetivo de la tesis, sino como elemento colateral que influye en nuestro enfoque. Aun así, la investigación realizada obtiene una valoración entre moderada y muy alta.

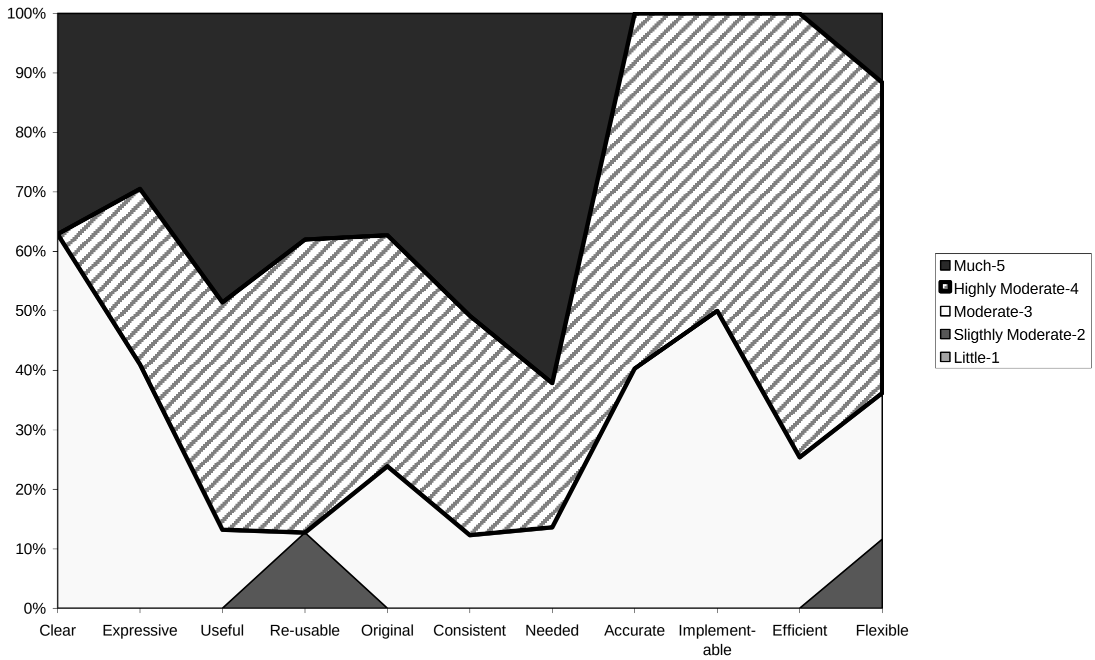
Por último, el trabajo apuntado sobre compartición de propiedades y estados muestra una satisfacción alta o muy alta para casi el 80% de los encuestados.

8. The solutions (extensions and modifications) presented in this thesis focused on ADAPTIVE LEARNING are



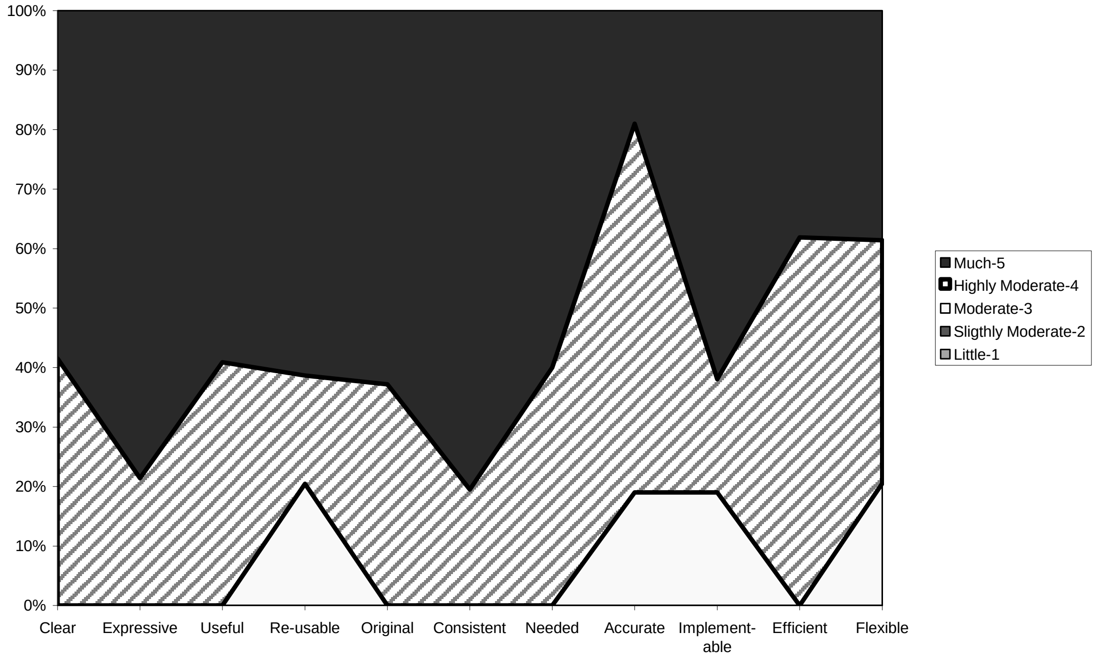
	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Clear	13,2	0,0	0,0	36,8	50,0	4,10	1,30
Expressive	0,0	0,0	0,0	**50,0**	**50,0**	4,50	0,50
Useful	0,0	0,0	0,0	**29,8**	**70,2**	4,70	0,46
Re-usable	0,0	0,0	15,0	13,3	71,7	4,57	0,74
Original	0,0	0,0	0,0	**55,9**	**44,1**	4,44	0,50
Consistent	0,0	0,0	0,0	**14,0**	**86,0**	4,86	0,35
Needed	0,0	0,0	0,0	**15,3**	**84,7**	4,85	0,36
Accurate	0,0	0,0	12,9	43,5	43,5	4,30	0,69
Implement-able	0,0	0,0	29,3	43,1	27,6	3,98	0,76
Efficient	0,0	0,0	0,0	**70,2**	**29,8**	4,30	0,46
Flexible	0,0	0,0	29,3	43,1	27,6	3,98	0,76
					Avg	4,42	0,63

Figure 53. Soluciones de aprendizaje adaptativo para diseñadores de aprendizaje



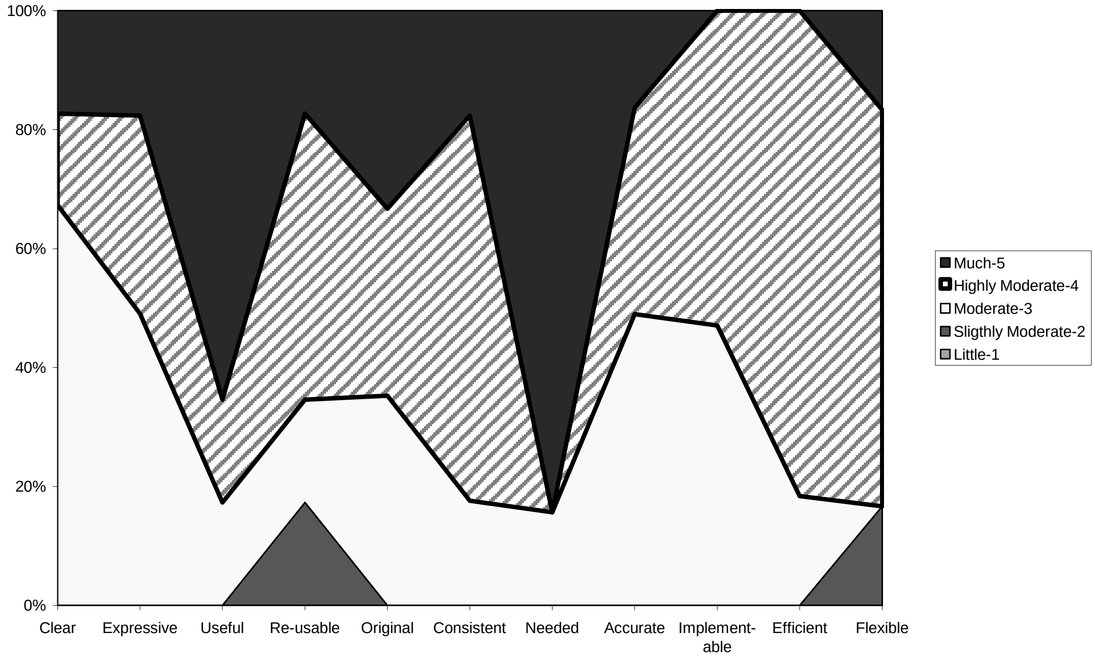
	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Clear	0,0	0,0	62,9	0,0	37,1	3,74	0,97
Expressive	0,0	0,0	41,0	**29,5**	**29,5**	3,89	0,84
Useful	0,0	0,0	13,2	**38,2**	**48,5**	4,35	0,71
Re-usable	0,0	**12,7**	0,0	**49,3**	**38,0**	4,13	0,94
Original	0,0	0,0	23,9	**38,8**	**37,3**	4,13	0,77
Consistent	0,0	0,0	12,3	**36,9**	**50,8**	4,39	0,70
Needed	0,0	0,0	13,6	**24,2**	**62,1**	4,48	0,73
Accurate	0,0	0,0	40,3	59,7	0,0	3,60	0,49
Implement-able	0,0	0,0	50,0	50,0	0,0	3,50	0,50
Efficient	0,0	0,0	25,4	**74,6**	0,0	3,75	0,44
Flexible	0,0	**11,6**	24,6	52,2	11,6	3,64	0,84
					Avg	3,96	0,72

Figure 54. Soluciones de aprendizaje adaptativo para profesores



	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Clear	0,0	0,0	0,0	**41,5**	**58,5**	4,59	0,50
Expressive	0,0	0,0	0,0	**21,4**	**78,6**	4,79	0,41
Useful	0,0	0,0	0,0	**40,9**	**59,1**	4,59	0,49
Re-usable	0,0	0,0	20,5	18,2	61,4	4,41	0,81
Original	0,0	0,0	0,0	**37,2**	**62,8**	4,63	0,49
Consistent	0,0	0,0	0,0	**19,5**	**80,5**	4,81	0,40
Needed	0,0	0,0	0,0	**40,0**	**60,0**	4,60	0,49
Accurate	0,0	0,0	19,0	61,9	19,0	4,00	0,62
Implement-able	0,0	0,0	19,0	19,0	61,9	4,43	0,79
Efficient	0,0	0,0	0,0	**61,9**	**38,1**	4,38	0,49
Flexible	0,0	0,0	20,5	40,9	38,6	4,18	0,75
					Avg	4,49	0,57

Figure 55. Soluciones de aprendizaje adaptativo para desarrolladores



	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Clear	0,0	0,0	67,3	15,4	17,3	3,50	0,78
Expressive	0,0	0,0	49,0	33,3	17,6	3,68	0,76
Useful	0,0	0,0	17,3	**17,3**	**65,4**	4,48	0,78
Re-usable	0,0	**17,3**	17,3	48,1	17,3	3,65	0,96
Original	0,0	0,0	35,3	31,4	33,3	3,98	0,83
Consistent	0,0	0,0	17,6	**64,7**	**17,6**	4,00	0,60
Needed	0,0	0,0	15,7	0,0	**84,3**	4,69	0,73
Accurate	0,0	0,0	49,0	34,7	16,3	3,67	0,74
Implement-able	0,0	0,0	47,1	52,9	0,0	3,53	0,50
Efficient	0,0	0,0	18,4	**81,6**	**0,0**	3,82	0,39
Flexible	0,0	**16,7**	0,0	**66,7**	**16,7**	3,84	0,90
					Avg	3,89	0,72

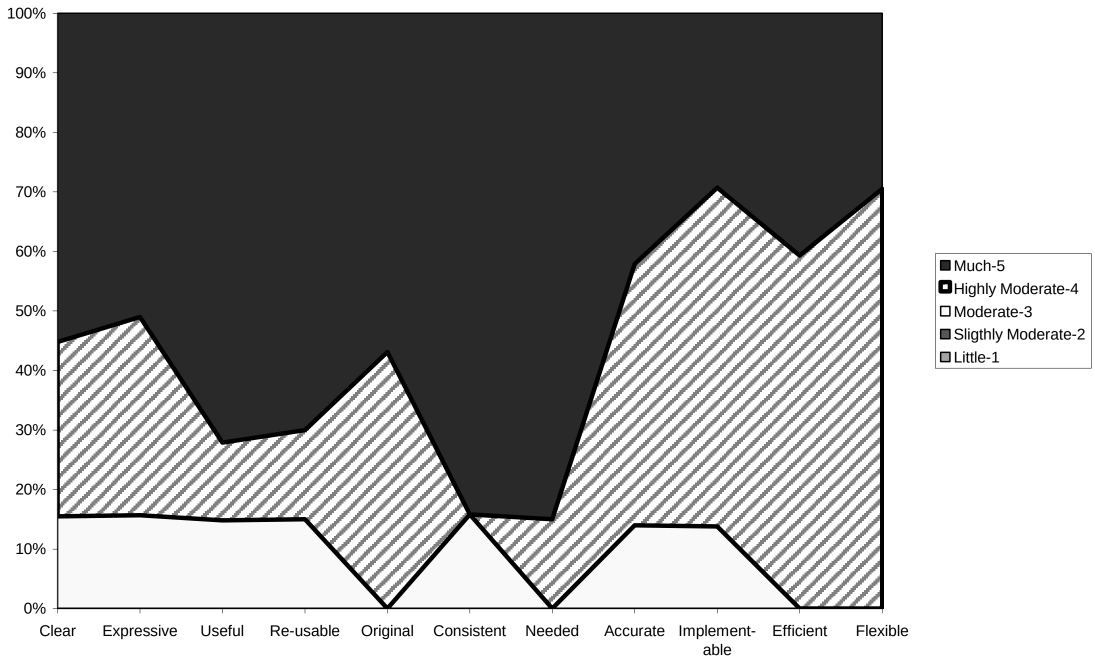
Figure 56. Soluciones de aprendizaje adaptativo para alumnos

De todos estos diagramas expresados en área porcentual sobre las soluciones aportadas para aprendizaje adaptativo (Figure 53, Figure 54, Figure 55, Figure 56) y distribuidas por público objetivo (diseñadores de aprendizaje, profesores, desarrolladores y alumnos) obtenemos varias conclusiones, según el punto de vista de los encuestados:

- Los cuatro colectivos se verían afectados de manera positiva y de forma alta o muy alta en términos generales
- Cabe destacar de forma alta o muy alta, cubriendo el 100%:

- o Para diseñadores de aprendizaje: Expresividad, utilidad, originalidad, consistencia, necesidad y eficiencia
  - o Para desarrolladores: Claridad, expresividad, utilidad, originalidad, consistencia, necesidad y eficiencia
- Para los profesores, lo más destacable es: Expresividad, utilidad, reutilización, originalidad, consistencia, necesidad y eficiencia, todo ello considerado como alto o muy alto por encima del 75%
- Para los alumnos: Utilidad, consistencia, necesidad, eficiencia y flexibilidad, como alto o muy alto por encima del 75%
- En general, ningún aspecto es considero bajo o muy bajo, salvo la claridad para diseñadores de aprendizaje (muy bajo para 13%), la reutilización y la flexibilidad para profesores y para estudiantes (bajo entre el 11% y el 17%)
- Si incluimos la calificación moderada, la media supera ampliamente el 80% de aceptación

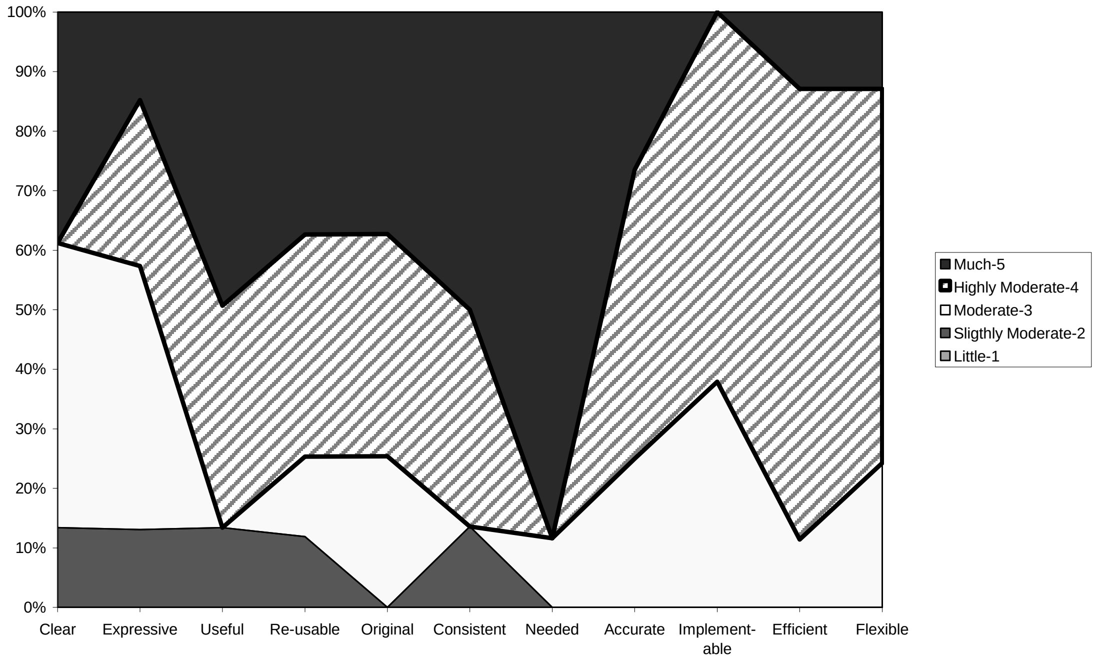
9. The solutions (extensions and modifications) presented in this thesis focused on INTEGRATION of Units of Learning are



	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Clear	0,0	0,0	15,5	29,3	55,2	4,40	0,74
Expressive	0,0	0,0	15,7	33,3	51,0	4,35	0,74
Useful	0,0	0,0	14,8	13,1	72,1	4,57	0,74
Re-usable	0,0	0,0	15,0	15,0	70,0	4,55	0,74

Original	0,0	0,0	0,0	**43,1**	**56,9**	4,57	0,50
Consistent	0,0	0,0	15,8	0,0	84,2	4,68	0,73
Needed	0,0	0,0	0,0	**15,0**	**85,0**	4,85	0,36
Accurate	0,0	0,0	14,0	43,9	42,1	4,28	0,70
Implement-able	0,0	0,0	13,8	56,9	29,3	4,16	0,64
Efficient	0,0	0,0	0,0	**59,3**	**40,7**	4,41	0,49
Flexible	0,0	0,0	0,0	**70,5**	**29,5**	4,30	0,46
					Avg	4,47	0,62

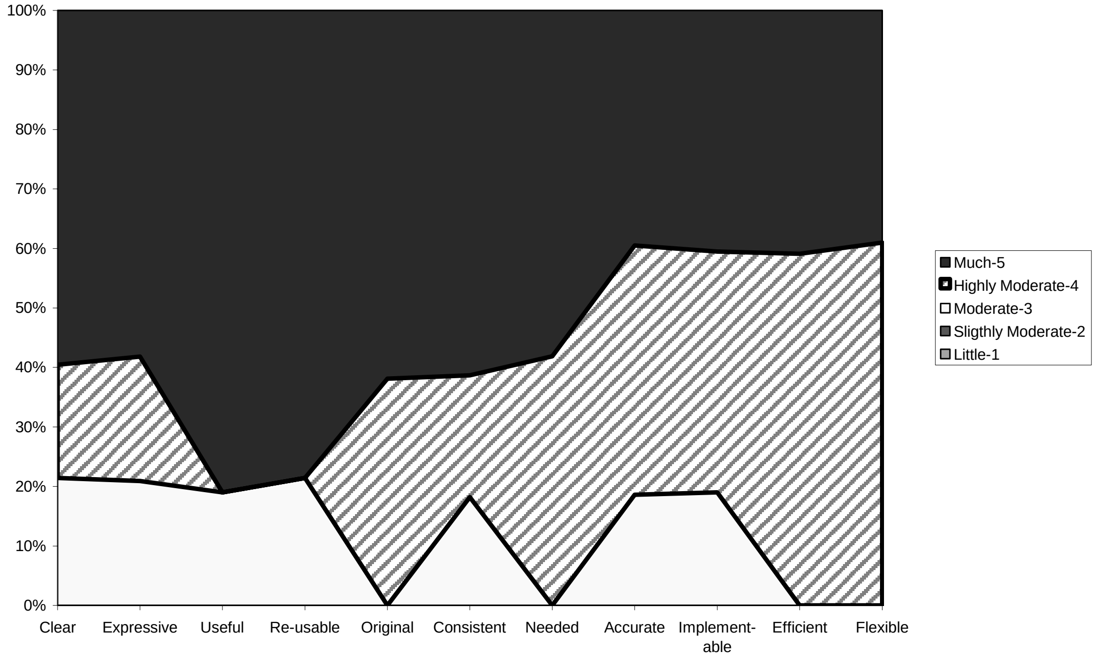
Figure 57. Soluciones de integración de Unidades de Aprendizaje para diseñadores de aprendizaje



	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Clear	0,0	**13,4**	47,8	0,0	38,8	3,64	1,13
Expressive	0,0	**13,1**	44,3	27,9	14,8	3,45	0,90
Useful	0,0	**13,4**	0,0	**37,3**	**49,3**	4,23	0,99
Re-usable	0,0	**11,9**	13,4	**37,3**	**37,3**	4,00	1,00
Original	0,0	0,0	25,4	**37,3**	**37,3**	4,12	0,79
Consistent	0,0	**13,6**	0,0	**36,4**	**50,0**	4,23	1,00
Needed	0,0	0,0	11,6	0,0	**88,4**	4,77	0,64
Accurate	0,0	0,0	25,0	48,5	26,5	4,02	0,72

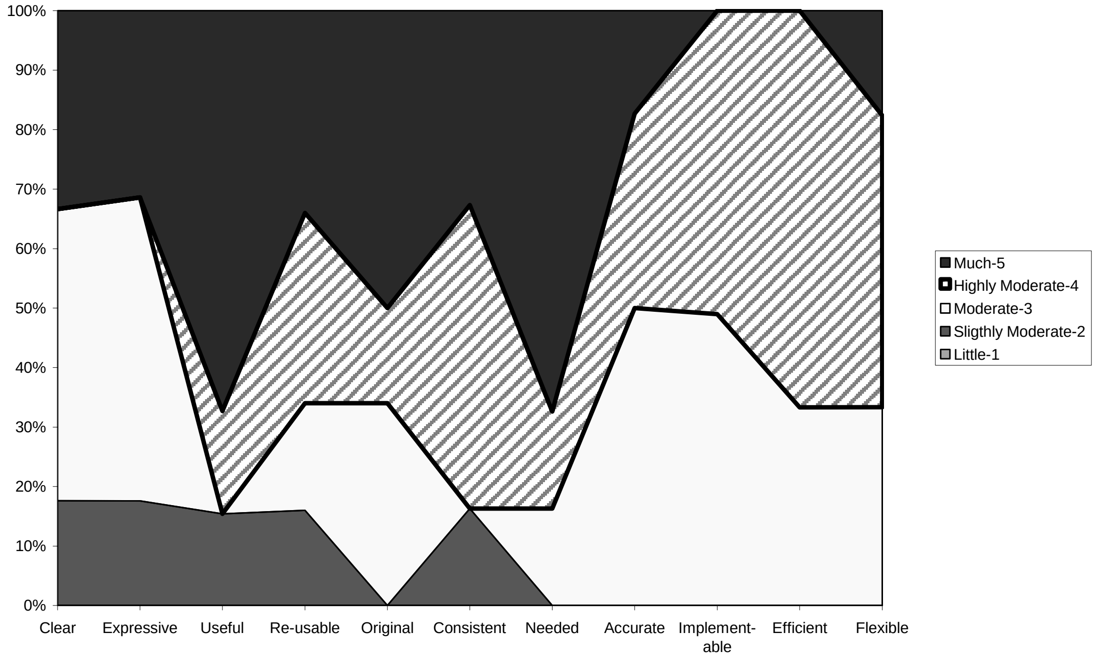
Implement-able	0,0	0,0	37,9	62,1	0,0	3,62	0,49
Efficient	0,0	0,0	11,4	**75,7**	**12,9**	4,02	0,50
Flexible	0,0	0,0	24,3	62,9	12,9	3,89	0,60
					Avg	4,00	0,80

Figure 58. Soluciones de integración de Unidades de Aprendizaje para profesores



	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Clear	0,0	0,0	21,4	**19,0**	**59,5**	4,38	0,82
Expressive	0,0	0,0	20,9	**20,9**	**58,1**	4,37	0,81
Useful	0,0	0,0	19,0	0,0	**81,0**	4,62	0,79
Re-usable	0,0	0,0	21,4	0,0	78,6	4,57	0,82
Original	0,0	0,0	0,0	**38,1**	**61,9**	4,62	0,49
Consistent	0,0	0,0	18,2	**20,5**	**61,4**	4,44	0,78
Needed	0,0	0,0	0,0	**41,9**	**58,1**	4,58	0,50
Accurate	0,0	0,0	18,6	**41,9**	**39,5**	4,21	0,74
Implement-able	0,0	0,0	19,0	**40,5**	**40,5**	4,22	0,74
Efficient	0,0	0,0	0,0	**59,1**	**40,9**	4,41	0,49
Flexible	0,0	0,0	0,0	**61,0**	**39,0**	4,39	0,49
					Avg	4,44	0,68

Figure 59. Soluciones de integración de Unidades de Aprendizaje para desarrolladores



	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Clear	0,0	**17,6**	49,0	0,0	33,3	3,49	1,13
Expressive	0,0	**17,6**	51,0	0,0	31,4	3,45	1,11
Useful	0,0	**15,4**	0,0	17,3	67,3	4,37	1,08
Re-usable	0,0	**16,0**	18,0	32,0	34,0	3,84	1,07
Original	0,0	0,0	34,0	16,0	50,0	4,16	0,91
Consistent	0,0	**16,3**	0,0	**51,0**	**32,7**	4,00	0,99
Needed	0,0	0,0	16,3	**16,3**	**67,3**	4,51	0,76
Accurate	0,0	0,0	50,0	32,7	17,3	3,67	0,76
Implement-able	0,0	0,0	49,0	51,0	0,0	3,51	0,50
Efficient	0,0	0,0	33,3	66,7	0,0	3,67	0,47
Flexible	0,0	0,0	33,3	49,0	17,6	3,84	0,70
					Avg	3,86	0,86

Figure 60. Soluciones de integración de Unidades de Aprendizaje para alumnos

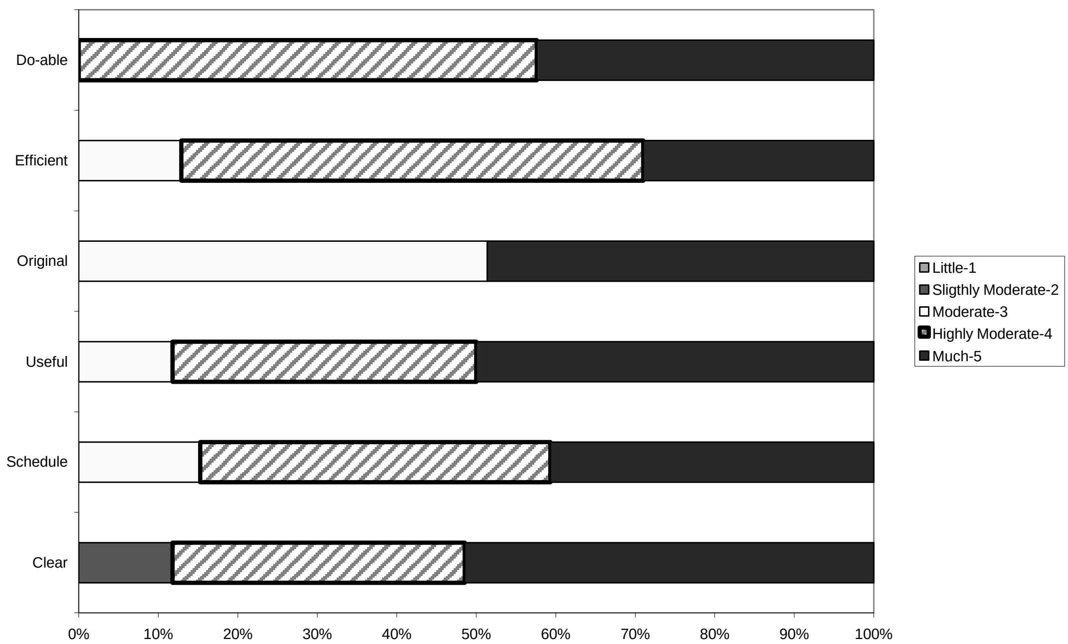
De todos estos diagramas expresados en área porcentual sobre las soluciones aportadas para aprendizaje adaptativo (Figure 57, Figure 58, Figure 59, Figure 60) y distribuidas por público objetivo (diseñadores de aprendizaje, profesores, desarrolladores y alumnos) obtenemos varias conclusiones, según el punto de vista de los encuestados:

- Los cuatro colectivos se verían afectados de manera positiva y de forma alta o muy alta en términos generales
- Cabe destacar de forma alta o muy alta, cubriendo el 100%:

- o Para diseñadores de aprendizaje: Originalidad, eficiencia y flexibilidad
- o Para profesores: Utilidad y consistencia
- o Para desarrolladores: Originalidad, necesidad, eficiencia y flexibilidad
- o Para estudiantes: Utilidad y consistencia

- Para los diseñadores de aprendizaje y para los desarrolladores, todos los aspectos son considerados de importancia alta o muy alta, por encima del 80% y del 79% por ciento, respectivamente
- Para profesores y estudiantes, todos los aspectos son considerados de importancia moderada, alta o muy alta, por encima del 80%
- Ningún aspecto es considerado muy bajo
- Para profesores y estudiantes, entre un 11% y un 17% consideran bajo: Claridad, expresividad, utilidad, reutilización y consistencia

10. The evaluation process is



	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5	Average	SD
Clear	0,0	11,8	0,0	36,8	51,5	4,28	0,96
Scheduled	0,0	0,0	15,3	**44,1**	**40,7**	4,26	0,71
Useful	0,0	0,0	11,8	38,2	50,0	4,38	0,69
Original	0,0	0,0	**51,4**	0,0	**48,6**	3,97	1,00
Efficient	0,0	0,0	12,9	58,1	29,0	4,16	0,63
Do-able	0,0	0,0	0,0	57,6	42,4	4,42	0,50
					Avg	4,25	0,75

Figure 61. Proceso de evaluación

El proceso de evaluación (Figure 61) es respaldado ampliamente en todas las categorías, resaltando por encima del 85% la claridad, la planificación, la utilidad, la eficiencia y la posibilidad de implementación. En cuanto a la originalidad del proceso, la mitad opina que es muy alta y la otra mitad que es moderada.

11. Does the solution and-or approach of this thesis show any important drawback?

El 98,4% opina que la solución no muestra ningún defecto importante. Los comentarios adicionales principales que ayudarían a completar la tesis son:

- Sugerencia de precisar de manera muy específica la definición de aprendizaje adaptativo, dado que según la disciplina desde la que se aborde, las connotaciones son muy distintas
- Sugerencia de dar una continuidad a la investigación para lograr una proyección de las soluciones propuestas
- Sugerencia de proporcionar una base más sólida sobre modelado de usuario
- Sugerencia de examinar la viabilidad económica asociada a las soluciones técnicas propuestas y su implicación con los modelos de código abierto (1,6% complementario)

12. Does the solution and-or approach of this thesis show any relevant contribution?

El 100% respalda la contribución relevante de la solución y/o de enfoque de la tesis al problema presentado. A modo de muestra, y englobando apreciaciones similares, algunos comentarios textuales a destacar son:

- *The solutions of this thesis show a relevant contribution in order to improve the learning flow, because these extension will allow to develope relevant activities involved in the learning process such as sharing information, monitoring activities, cheking time per session, notifying relevant issues, user grouping, etc*
- *It should be a very important contribution to the IMS-LD specification if it extends the current specification*
- *The new contributions that I appreciate more are as follows: (1) improvement in the learning flow (e.g., new conditional and iterative structures), (2) extending monitoring features (e.g., role tracking in groups), (3) greater control in UoL (e.g., assign an activity to a particular user), (4) group*

management (e.g., role tracking within groups), (5) supporting run-time modifications (e.g., new resources can be included at run-time). Above all I would encourage further developments to improve run-time adaptive features that clearly support the evolving nature of learning design

- *Las soluciones propuestas de mejora o complemento de IMS-LD presentadas en la tesis contribuyen de manera importante a la tendencia actual en educacion de pasar a una "educacion para todos" que presenta a los estudiantes una secuencia fija y predeterminada de tareas de aprendizaje, a implementar una educacion "solo para mi" que ofrece un itinerario curricular personalizado a cada alumno. Este proceso requiere emparejar tareas de aprendizajes con alumnos, y la asignacion de una actividad apropiada al alumno. Otra aportacion importante es el reconocimiento de facilitar la utilizacion y uso de la herramienta a grupos con conocimientos tecnicos limitados, como profesores o disenadores instruccionales*
- *Creo personalmente, que después de años de trabajo en el sector se muestran deficiencias relevantes de una especificación que tiene un gran potencial y que sobre todo ayuda a la estandarización de procesos de aprendizaje. Las soluciones mostradas son un camino ambicioso por el que se ha de luchar para conseguir la viabilidad, ya que nos ofrecen un abanico de posibilidades inmensas*

13. Would you like to use these types of solution in a system-specification-authoring tool?

Del mismo modo, el 100% se muestra partidario de utilizar este tipo de solución en un sistema, especificación o herramienta de autoría.

14. In the future, would you participate in a funded project supporting this contribution?

Por ultimo, el 90,4% reafirma su respaldo a la tesis indicando su disposición a participar en un proyecto que la utilizara de alguna manera (por ejemplo, como enfoque, fundamento o desarrollo).

Todos estos comentarios y sugerencias (preguntas 11 y 12), así como la lectura que arrojan los resultados estadísticos, serán tenidos en fases futuras de la investigación, dado su gran interés y la alta cualificación de los profesionales que amablemente los proporcionan, ya que entendemos que podrían contribuir a enriquecer y hacer más preciso el enfoque y/o las soluciones aportadas, así como el impacto de las mismas.

8. Conclusiones generales

Aunque el Nivel A de IMS Learning Design proporciona el esqueleto y la base de una unidad de aprendizaje con su itinerario formativo, el Nivel C y sobre todo el Nivel B proveen de los recursos más potentes y versátiles. Mientras que el Nivel A define el 80% de una UoL a nivel estructural, los Niveles B y C definen el 80% de la potencialidad. Con la adecuada combinación de los elementos descritos se pueden abordar retos actuales de la enseñanza como los enunciados (active learning, collaborative learning, adaptive learning, runtime tracking, etc).

IMS-LD se beneficiaría de una reestructuración y de una modificación de ciertos elementos de modelado y arquitectura, así como la incorporación de otros nuevos, para mejorar una expresividad pedagógica y una capacidad de integración con otros sistemas de aprendizaje y estándares eLearning, si se pretenden alcanzar dos de los objetivos principales establecidos de base en la definición de esta especificación: La personalización del proceso de aprendizaje y la interoperabilidad e integración real de unidades de aprendizaje. Aun así, es cierto que la implantación de la especificación se vería claramente mejorada si existieran unas herramientas de más alto nivel (preferiblemente con planteamiento visual) que permitieran un modelado sencillo por parte de los usuarios finales reales de este tipo de especificaciones, como son los profesores, los creadores de contenido y los pedagogos/didactas que diseñan la experiencia de aprendizaje. Este punto, no obstante, es ajeno a la especificación y afecta a la interpretación que de la misma realizan los grupos de investigación y compañías que desarrollan soluciones de autoría.

8.1 . Modelado

- IMS-LD maneja una metáfora incómoda de modelado. Aunque la metáfora que permite definir y relacionar los distintos componentes es clara desde un punto de vista conceptual, no lo es tanto cuando se intenta implementar un ejemplo concreto. La división entre `learning design`, `plays`, `acts` y `activity structures`, por ejemplo, proporciona una relación en cascada que, siendo versátil para crear escenarios complejos, dista mucho de los escenarios sencillos sujetos habitualmente a cursos específicos. Esta subordinación obligatoria de elementos añade confusión. Otro ejemplo sería el que muestra las referencias consecutivas que especifican un servicio y que necesitan definición en la actividad, el entorno, el servicio en sí, el recurso y el fichero que lo ejecuta

- La notación IMS-LD es difícil de concretar ya que sigue los parámetros habituales XML, algo no trivial incluso para programadores, pero define una estructura propia de manifiesto Content Packaging con una extensión de la etiqueta organizations, que genera un manifiesto Learning Design extenso y farragoso de seguir. Además, las etiquetas y definición de elementos es muy detallada lo que hace laborioso realizar labores sencillas, como una simple definición de variables, por ejemplo. La programación, por tanto, se vuelve complicada, a pesar de su sencillez
- Las estructuras de programación proporcionadas son muy básicas (condición simple, aritmética simple, edición y visualización de variables, gestión de visibilidad de actividades y capas DIV), además de utilizar una sintaxis extensa, lo que dificulta la visibilidad. Una extensión de estas estructuras y una modificación de su sintaxis aumentaría exponencialmente la potencialidad de la especificación en la flexibilidad pedagógica que busca

8.2 . Comunicación, interoperabilidad e integración de Unidades de Aprendizaje

- Para lograr la interoperabilidad e integración de Unidades de Aprendizaje con otras especificaciones y módulos externos a IMS-LD se proponen tres soluciones. La primera se centra en la vínculación externa del objeto de aprendizaje sin establecimiento de ninguna comunicación con la UoL. Esta solución anula cualquier intercambio entre ambos entes más allá del contacto nominal. La segunda solución se centra en la inclusión de un paquete de información dentro del paquete IMS-LD, convirtiéndolo en parte integrante del flujo de aprendizaje dentro de la UoL, pero sin compartición aún de variables comunes
- La tercera solución se centra en la creación de una capa de comunicación que permita el intercambio efectivo de información, como variables y valores de variables y estado, lo que posibilitará interactuar IMS-LD con un módulo externo y viceversa
- La existencia de una capa de comunicación y el desarrollo de la interoperabilidad facilitaría la relación de IMS-LD con módulos y aplicaciones desarrollados con otras especificaciones (IMS Content Packaging, Scorm) otros lenguajes (PHP, Java) y con otros sistemas (Lams, Moodle). Uno de estos grupos lo constituyen los campus virtuales (*Learning Management System –LMS-* o *Virtual Learning Environment –VLE-*) desarrollado bajo cánones de IMS-LD o compatible con Unidades de Aprendizaje creadas en IMS-LD, con posibilidad de intercambio (importación y exportación) y ejecución y una capa de servicios (foros, chats,

administración de usuarios, seguimiento de expediente, trabajo colaborativo, servicio de noticias o correo) que permita interactuar con ellas

- Del mismo modo, junto con la interoperabilidad, el intercambio de paquetes logrará la re-utilización de Unidades de Aprendizaje previamente modeladas, bien para su re-definición, bien para su compilación e inclusión en unidades mayores. Este intercambio de paquetes puede servir de base para el desarrollo de plantillas o patrones de diseños instructivos interoperables, lo que facilitaría la creación y utilización de la especificación por todas las comunidades de práctica involucradas y no únicamente por el sector más tecnológico
- No existe ninguna posibilidad de realizar conexiones con bases de datos o ficheros externos que impliquen la importanción/exportación/compartición de datos. De esta manera, la integración con LMSs o cualquier otro tipo de aplicaciones que pudieran requerir el intercambio efectivo de variables, propiedades, estados o paquetes de información con IMS-LD se vuelve imposible
- Por último, la incorporación de elementos que puedan visualizar y modificar el learning design durante la ejecución y que, de hecho, rompa la división actual entre tiempo de diseño y tiempo de ejecución, pasando de un lenguaje compilado a uno intérprete, impulsaría ampliamente la adaptación interactiva del itinerario de aprendizaje. Este apartado corresponde a la interpretación que los desarrolladores de herramientas y motores de ejecución realizan y no al propio diseño de la especificación, ajeno a esta problemática

8.3 . Editores visuales de alto nivel

Aunque fuera del núcleo principal de investigación de esta tesis, el estudio de los editores también aporta unas pautas que consideramos interesante reflejar aquí:

- Son necesarios editores o herramientas de autoría de alto nivel con un diseño gráfico y un diseño de información más centrados en el usuario final y menos en las especificaciones y requerimientos técnicos. Es cierto que en este último año 2006-2007 hemos pasado de no contar con ninguna aplicación basada en IMS-LD a tener más de una docena centradas o en torno a IMS-LD, lo que supone un gran avance; pero el punto en común de todas ellas es la realización técnica de lo que la especificación define, no la utilización sencilla por usuarios reales. Es decir, con las aplicaciones informáticas actuales se pueden construir Unidades de Aprendizaje, mayormente sencillas o en nivel A, pero el grado de conocimientos técnicos necesarios es alto y la facilidad de uso de los interfaces es escasa. Por otra parte, la edición de un Nivel B/C es extremadamente laboriosa y encriptada, resultando más fácil abordar la creación de este tipo de UoLs directamente con un editor plano de propósito general (tipo bloc de notas) o con un editor tipo XML. Esta edición más técnica, vuelve a restringir las posibilidades de utilización directa por parte de la comunidad virtual de usuarios no técnicos (profesores, proveedores de contenidos, estudiantes no tecnológicos…) y también dificulta una implementación sencilla y ágil incluso por parte de usuarios técnicos experimentados (diseñadores de aprendizaje, desarrolladores). La edición debería estar lo más independiente posible de la estructura técnica subyacente, incluso resultando opaca la división por niveles o la sintaxis. Un enfoque más visual, con editores de alto nivel, y una metáfora de utilización más próxima a la realidad cotidiana de profesores y alumnos no técnicos, supone un avance necesario para el uso correcto, extendido y real de la especificación con todo su potencial. Esto no quita que sigan existiendo los editores técnicos y de propósito general para aquellos conocedores de la notación y de IMS-LD en profundidad, editores de los que ya disponemos y que ya han sido mencionados (CopperAuthor, Reload LD Editor, Cosmos…)

- Es necesario un interfaz con un mayor grado de usabilidad, con una metáfora de aplicación y un sistema de diseño gráfico *drag&drop*, además de una ayuda contextual y bien documentada, en vez de un sistema de solapas y etiquetas para rellenar sin mayor soporte ni información y sin una conexión educativamente metodológica entre ellas, que es lo que existe en la actualidad. Se necesita, por tanto, una evolución que permita trabajar de manera conjunta con el manifiesto, con los recursos y con los ficheros externos XHTML relacionados, mostrando las dependencias entre ellos y un sistema sencillo de edición y configuración de las propiedades

8.4 . Extensión y modificación de IMS Learning Design

Con el objeto de mejorar la expresividad pedagógica de IMS-LD, y centrados en el aprendizaje adaptativo y en la integración de Unidades de Aprendizaje, proponemos la modificación e incorporación de estructuras de modelado y arquitectura. En concreto:

a) Desarrollo de nuevas estructuras y elementos de modelado de propósito general y modificación de la definición de estructuras existentes. Estos elementos y estructuras serán utilizados como parte de otros desarrollados de manera específica para sustentar procesos de aprendizaje adaptativo. Constituyen, pues, un conjunto básico para ser utilizado directamente o como parte de un enfoque de personalización

b) Desarrollo de estructuras específicas para aprendizaje adaptativo. Ciertos procesos no pueden ser abordados con la única definición de estructuras generales y necesitan la definición adicional de estructuras y elementos específicos. De esta manera se logra conseguir el objetivo de adaptación de una manera, y se describe de una manera más directa

c) Modificación y desarrollo de estructuras que posibiliten y favorezcan una integración e interoperabilidad satisfactorias. La comunicación y compartición de información sobre usuarios, actividades y procesos con aplicaciones externas (por ejemplo, sistemas de aprendizaje) es algo no implementado de manera efectiva aún y garantiza la utilización de esta especificación en entornos reales de aprendizaje

Con todo ello se define cómo puede dicha especificación ser modificada o extendida para lograr una respuesta suficiente y optimizada que proporcione un mayor grado de identificación entre objetivos educativos, recursos tecnológicos y resultados de aprendizaje en la práctica educativa real. En definitiva, una mejora de la expresividad pedagógica.

Por lo tanto, propondremos las modificaciones y extensiones de las estructuras existentes así como el desarrollo de cuantas nuevas estructuras sean necesarias para dotar a la especificación de una respuesta educativa eficaz mediante los recursos y estructuras tecnológicas adecuadas.

En concreto, desarrollaremos extensiones y modificaciones basadas en dos categorías:

c) Modelado: Que aborda el *information model* y aspectos específicos de código, con modificación de determinadas estructuras y modificación de sintaxis, de gramática y funcionalidad de otras. Por ejemplo: estructuras condicionales, global elements y acceso a propiedades, servicio de monitorización

d) Arquitectura: Que se centra en el aspecto funcional de la especificación, en concreto, la comunicación e integración de IMS-LD con otros sistemas y plataformas externas. Por ejemplo: Grabación y recuperación de datos, enlace con bases de datos, modificación dinámica del *learning design*

Se busca la máxima integración y la máxima funcionalidad con el mínimo impacto dentro de la estructura posible, es decir, respetar al máximo la especificación tal y como es con sus definiciones actuales y realizar los cambios e incorporaciones mínimos necesarios para completarla y extendarla acorde con nuestro análisis. En la siguiente tabla se muestran los distintos elementos de análisis y crítica (Table 12):

Table 12. Resumen de los elementos de análisis y crítica de la especificación IMS-LD

ID	Elementos de análisis y crítica
Modelado	
[M.01]	IMS-LD maneja una sintaxis y una gramática incómodas de modelado. La división entre learning design, plays, acts y activity structures proporciona una subordinación obligatoria de elementos que añade confusión
[M.02]	Las referencias consecutivas que definen un servicio generan demasiados pasos intermedios para algo tan sencillo
[M.03]	El diseño del learning design en sí es lógico y portable
[M.04]	La notación IMS-LD es difícil de concretar. Se genera un manifiesto Learning Design extenso y farragoso de seguir. El modelado o programación, por tanto, se vuelven complicados, a pesar de su sencillez conceptual
[M.05]	Las estructuras de programación proporcionadas son muy básicas (condición simple, aritmética simple, edición y visualización de variables, gestión de visibilidad de actividades y capas DIV)
[M.06]	No se contempla el tiempo relativo de ejecución desde el momento de comienzo de la ejecución de una actividad por parte del usuario o desde cualquier otro momento de sincronización. Tampoco se contempla la referencia a tiempo absoluto
[M.07]	El servicio de notificación del Nivel C contempla únicamente un envío de correo o una realización de una actividad de aprendizaje o soporte. Se encuentra infrautilizado
[M.08]	La definición y uso de propiedades y enlaces entre los diversos ficheros XML que permiten trabajar con los elementos globales son confusos
[M.09]	La relación entre capas DIV y la gestión de la propiedad de visibilidad podría simplificarse para lograr una mayor claridad de definición y seguimiento
[M.10]	Existe una única manera de establecer condiciones, mediante la

	estructura `if-then-else` y anidaciones de la misma
[M.11]	Las expresiones que evalúa la estructura `if-then-else` son, no obstante, las habituales en este tipo de estructuras
[M.12]	No existe ninguna estructura de iteración en ninguno de los elementos básicos (actividad, estructura de actividades, actos, obras)
[M.13]	No existen puntos de entrada o sincronización dentro de la estructura del manifiesto
[M.14]	No se puede realizar la asignación concreta de una actividad específica a un usuario seleccionado
[M.15]	No existe la creación de grupos de manera selectiva dentro de la instancia
[M.16]	No existe una estructura de trabajo colaborativo que no haya sido expresamente modelada con los recursos básicos existentes
[M.17]	El servicio de monitorización no contempla la monitorización grupal
[M.18]	No existe ningún sistema de comunicación entre usuarios/roles más allá de la monitorización uno a uno mediante la visualización de propiedades ajenas
Arquitectura	
[A.01]	Para lograr la interoperabilidad con otras especificaciones y módulos externos a IMS-LD es necesario un sistema o capa de comunicación que permita el intercambio efectivo de información, como variables y valores de variables y estado, lo que permitirá interactuar desde IMS-LD en el módulo externo y viceversa
[A.02]	IMS-LD no permite la grabación ni la recuperación de datos en y desde ficheros externos de ningún formato. Del mismo modo, las conexiones con bases de datos externas o con módulos programados en otros lenguajes no están contempladas dentro de la especificación
[A.03]	Se hace preciso el desarrollo de una capa de comunicación o *dispatcher* para lograr la integración de módulos externos con IMS-LD

[A.04]	No hay posibilidad de modificar el esqueleto, el método, la definición de roles o cualquier otro elemento estructural en la fase de ejecución o *run-time*
Herramienta	
[H.01]	La ejecución actual de las herramientas bajo IMS-LD no soporta la modificación de una Unidad de Aprendizaje en tiempo real, definiendo de facto un divorcio entre el tiempo de diseño y el tiempo de ejecución
[H.02]	Se necesita una evolución de los editores técnicos
[H.03]	Se necesita impulsar la creación de patrones de aprendizaje y el desarrollo de editores visuales
[H.04]	El intercambio de paquetes logrará la re-utilización de Unidades de Aprendizaje previamente modeladas, bien para su re-definición, bien para su compilación e inclusión en unidades mayores
[H.05]	Son necesarios editores o herramientas de autoría con un diseño gráfico y de información más centrados en el usuario final y menos en los logros técnicos
[H.06]	La importación y la exportación al 100% representan todavía un obstáculo y constituyen por tanto un gran paso para lograr el intercambio efectivo de paquetes de información

Basándonos en el detalle de la tabla anterior en la siguiente tabla (Table 20) concretamos las soluciones específicas que consitituyen la contribución base de esta tesis (detalle de cada una en la Sección 6). Remarcar que no todos los elementos de análisis y crítica reciben tratamiento o sugerencias de modificación o inclusión, dado que algunas de ellas se agrupan en torno a críticas más generales o no requieren de una solución concreta dado su carácter más generalista. Del mismo modo, ciertos elementos conllevan varias soluciones en paralelo y algunas soluciones abarcan varios elementos. Se deduce por tanto el carácter altamente entrelazado que tanto los elementos de análisis como las soluciones desarrolladas mantienen.

Table 20. Soluciones desarrolladas

ID	Elementos de análisis y crítica. Soluciones desarrolladas
[M.05] [M.10] [M.11]	Las estructuras de programación proporcionadas son muy básicas (condición simple, aritmética simple, edición y uso de variables, gestión de visibilidad de actividades y capas DIV)
[Ext.01a]	Estructura condicional de tipo case
[Ext.01b]	Estructura condicional de tipo case con rangos automáticos
[Ext.02]	Bucle de condición inicial tipo while
[Ext.03]	Bucle tipo for-next
[Ext.04]	Modificación de la estructura de cálculo <calculate>
[M.06]	No se contempla el tiempo relativo de ejecución desde el momento de comienzo de la ejecución de una actividad por parte del usuario o desde cualquier otro momento de sincronización. Tampoco se contempla la referencia a tiempo absoluto
[Ext.05]	Se modifica la referencia al tiempo relativo y se añade la referencia a tiempo absoluto
[M.07]	El servicio de notificación del Nivel C contempla únicamente un envío de correo o una realización de una actividad de aprendizaje o soporte. Se encuentra infrautilizada
[Ext.06]	Extensión del servicio de notificación para que admita otras acciones aparte de sendmail y actividad y para que pueda ser llamado desde otras estructuras y no únicamente desde la parte <on-completion> de una learning activity
[M.08]	La definición y uso de propiedades y enlaces entre los diversos ficheros XML que permiten trabajar con los elementos globales son confusos
[Ext.07]	Modificación de la sintaxis, definición y utilización de los elementos view-property y set-property, así como de la definición de las variables que los

	utilizan
[M.09]	La relación entre capas DIV y la gestión de la propiedad de visibilidad podría simplificarse para lograr una mayor claridad de definición y seguimiento
[Ext.08]	Por defecto, asumir como estado hide la propiedad de visibilidad de capas para conseguir que la estructura condicional que las gestiona se simplifique
[M.12]	No existe ningún soporte de iteración en ninguna de las estructuras básicas de la metáfora (actividad de aprendizaje, actividad de soporte, estructura de actividades, actos, obras)
[Ext.09]	Extender la sintaxis actual de cada elemento con un parámetro iteration que define una iteración numérica (tipo for) y/o una iteración por condición de entrada y repetición (tipo while)
[M.13]	No existen puntos de entrada o sincronización dentro de la estructura del manifiesto
[Ext.10]	Incorporar un elemento GOTO que permita la redirección directa del flujo de aprendizaje
[M.14]	No se puede realizar la asignación concreta de una actividad específica a un usuario seleccionado
[Ext.11]	Incorporar un elemento ASSIGN-ACTIVITY-TO-USER que permite emparejar usuarios, grupos y roles con actividades de aprendizaje y estructuras de actividades
[Ext.12]	Incorporar un elemento ASSIGN-USER-TO-ACTIVITY que permite emparejar usuarios, grupos y roles con actividades de aprendizaje y estructuras de actividades
[Ext.13]	Incorporar un elemento SWITCH-ACTIVITY que permite activar y desactivar actividades y estructuras de actividades
[M.15]	No existe la creación de grupos de manera selectiva dentro de la instancia

[Ext.14]	Incorporación de un elemento CREATE-GROUP que permite agrupar usuarios dentro de un mismo rol

[M.17]	El servicio de monitorización no contempla la monitorización grupal
[Ext.15]	Extensión del servicio de monitorización para seguimiento de roles y grupos

[A.02]	IMS-LD no permite la grabación ni la recuperación de datos en y desde ficheros externos de ningún formato. Del mismo modo, las conexiones con bases de datos externas o con módulos programados en otros lenguajes no están contempladas dentro de la especificación
[Ext.16] [Ext.17] [Ext.18]	Incorporación de un elemento EXPORT y de un elemento IMPORT que gestionarán un fichero de tipo parametrizadao (tipo TXT, por ejemplo) declarado en una propiedad de un tipo nuevo FILE-IO
[Ext.19] [Ext.20] [Ext.21]	Incorporación de un elemento FROM-DB y de un elemento TO-DB que permiten recuperar y almacenar información de una base de datos en formato MySQL (en principio). La conexión de define en una propiedad de un tipo nuevo DATABASE

[A.04]	No hay posibilidad de modificar el esqueleto, el método, la definición de roles o cualquier otro elemento estructural en la fase de ejecución o run-time
[Ext.22] [Ext.23] [Ext.24] [Ext.25]	Se incorporan dos pares de *global elements* a) *view-IMS-LD* y *set-IMS-LD*, b) *view-resources* y *set-resources*, permitiendo visualizar y modificar la estructura del *learning design* y los recursos asignados en tiempo de ejecución

8.5 . **Futuro de investigación inmediato**

Las soluciones aquí desarrolladas servirán de base para implementación dentro de diversos proyectos europeos y nacionales detallados en la introducción de esta tesis. En concreto:

EU4ALL (www.eu4all-project.eu, 2006-2009, FP6 Europa), Integrated Project (IP) centrado en aprendizaje online para discapacitados. El subproyecto SP5 trabaja con IMS-LD y estructuras y escenarios de personalización basadas en guías pedagógicas y psicológicas.

Grapple (sin web todavía, 2008-2011, FP7 Europa), proyecto recientemente aprobado en FP7, que desarrollará un sistema de aprendizaje adaptativo y que contará con un paquete de trabajo centrado en IMS-LD y personalización. En este proyecto se contempla la discusión de extensiones y modificaciones de IMS-LD con el organismo de estandarización IMS

SUMA (sin web todavía, 2007-2009, Plan Avanza España), proyecto nacional que integra diversos servicios dentro de la capa OKI, entre ellos extensiones de Moodle y mapeo con IMS-LD. El subproyecto SP4 se centra en sistemas adaptativos.

Paralelamente, y plenamente relacionados con estos proyectos en su mayoría, se tendrán en cuenta tanto los resultados de la evaluación (Sección 7.2), como lo comentarios aportados durante la pre-defensa y la defensa de la tesis en futuras líneas de investigación. En concreto:

1. IMS-LD utiliza un modelo concurrente elemental de programación para sistematizar el flujo de actividades a realizar por los distintos roles. Otro enfoque posible al problema de la tesis hubiera sido partir de este modelo de concurrencia para evolucionarlo o complementarlo y así mejorar el rendimiento de la especificación
2. La solución planteada busca modificar o extender la especificación existente. En línea con una de las tendencias actuales sobre extensión de lenguajes de programación, otro enfoque sería el uso del *literate programming*, basado en la anotación sobre el lenguaje de programación y no en su modificación desde la estructura propia. De esta manera, se puede enriquecer un lenguaje sin necesidad de tocar su núcleo, algo complicado en desarrollos tan participativos y laboriosos como es la generación de especificaciones
    1. Tanto la programación concurrente [238-240] como el uso de *literate programming* [241; 242] son dos enfoques muy interesantes que pueden proporcionar distintas soluciones al problema planteado en esta tesis. Al ser ambos muy diferentes de la línea de trabajo llevada hasta ahora y, precisamente, reconociendo su interés, se estudiarán como políticas complementarias en una siguiente fase de investigación. De hecho, una línea ya abordada es la representación de mecanismos de coordinación en

IMS-LD. Como material complementario, incluimos un artículo en proceso de publicación en los anexos, Sección 9.3.

3. A lo largo de la investigación sobre aprendizaje adaptativo se estructuran dos bloques de tipos de adaptación. El *feedback* adaptativo tiene gran importancia en los sistemas actuales de aprendizaje online, al mismo nivel que el flujo de aprendizaje, el contenido o el interfaz. Desde un punto de vista funcional y didáctico el *feedback* adaptativo es de gran importancia y representa una tendencia en alza bien justificada que necesita una investigación más específica

4. La personalización de flujo y contenido con criterios de accesibilidad es una pauta recién abierta y con gran futuro de investigación que debe verse tratada como paso siguiente lógico de este trabajo. Cuando las herramientas y los visualizadores lo permitan, el estudio sobre accesibilidad debería extenderse al interfaz

5. La integración de IMS-LD con otros estándares y LMSs se considera vital para la adopción de esta especificación y el beneficio mutuo de los sistemas considerados en la tesis. Se necesitan unos modelos claros de migración para mapear notaciones y especificaciones y lograr una interoperabilidad eficaz

6. Lograr una herramienta visual que permita una edición de alto nivel contribuirá a la adopción del estándar y su utilización efectiva en las comunidades de práctica objetivo

7. El modelo conceptual de aprendizaje adaptativo necesita una mayor elaboración. Se beneficiaría de la inclusión de más roles e inputs, la concreción del papel de los grupos intra e inter roles, y del establecimiento de una clara relación entre todos estos factores que permita la formulación de una guía clara de adopción. Al mismo tiempo, el conjunto de reglas debe ser descrito y parametrizado, acorde con los demás factores. Todo de una manera genérica que pueda ser implementada en diversos LMSs y especificaciones que lo permitan en la actualidad o que, al menos, faciliten su incorporación como desarrollo adicional externo, en caso contrario

Conclusions

Level A of IMS-LD provides the basic skeleton and a general framework to work with Units of Learning. It makes the 80% of the whole structure. Level C, and above all, Level B, provide both the spec with stronger and more versatile resources. These two upper levels are the actual responsible means to model some of the current learning and teaching challenges (i.e., active learning, collaborative learning, adaptive learning, runtime tracking).

IMS-LD will benefit from a re-structure and modification of several elements focused on modeling and architecture. It will also improve the overall pedagogical expressiveness, along with specific features on adaptation of learning processes and integration with other specifications, LMSs and learning resources. These are two main objectives of the specification: personalised learning and interoperability. At the same time, IMS-LD would increase the implementation and a wider support if one or several high-level visual authoring tools are developed. Nevertheless, this issue is out of the scope of this thesis and it deals with research groups and companies working on the adoption of IMS-LD.

We depict our conclusions within the same two main blocks that we have used to carry out the analysis: modeling (with a special focus on adaptation) and integration. Out of our solution, we also provide a brief note about authoring tools.

8.6 . Modeling

With regards to general modeling and modeling focused on adaptive learning we conclude that IMS-LD shows a metaphor difficult to understand. It is not as much to say that people do not understand what a theatre is or how a play is performed. The key issue comes when a teacher needs to translate this well-know structure into specific pedagogical resources and features. This translation process turns not to be so obvious. The conceptual model is clear: play, acts, roles, role-parts, and so on. But all of them, interlaced in a whole structure of learning, become complex. Even the simplest scenario requires some knowledge of the specification in a technical way. And this is far from being user-friendly, moreover when the usual target people consists of non-technical profiles.

The notation itself follows a usual XML Schema and the definition of the several elements and components of the spec can turn too complex, even for skilled programmers. The description of activities, activity structures, environments, and etcetera, and the long cascade of relationships amongst them, makes a difficult-to-trace chain out of a simple scenario. Not to mention when several roles are involved, when some components of Level B are used or when adaptive processes are required. The programming structure is quite easy, but the sum of elements, components and metaphor, make it hard difficult to implement.

The programming components provided by IMS-LD are quite simple (i.e., simple condition, based arithmetic, visualization of variables, visibility, DIV layers, and etcetera). On the other side, their syntax is long, which hinders the rationale of the modeling process itself.

8.7 . Communication, interoperability, integration of Units of Learning

We study three ways of communication: 1) simple link between parts, 2) embedded information packages with no information exchange, and 3) full communication of information packages, sharing variables and states. This third solution becomes the most effective one. It implies the development of a communication layer that deals with effective bi-directional exchange of data between information packages. Furthermore, this solution allows for the communication and sharing of services, along with variables, values and states, between IMS-LD and any outside counterpart, i.e., other specifications (SCORM), languages (PHP, Java, Action Script), and LMSs (LAMS, Moodle, .LRN).

Should this exchange actually happens, it will encourage the re-use of information packages in different contexts, and the development of templates, fostering the re-purpose of Units of Learning within and amongst the several communities of practice (target groups) involved in IMS-LD, beyond the very only technical niche.

In the same line, exportation and importation of Units of Learning is not developed so far; neither does any connection with a database. Once more, no information exchange with other entities is possible so far.

The current two-step working process that makes two isolated parts out of design-time and run-time, makes IMS-LD to be compiled and not interpreted. This distinction stops an on-the-fly visualisation and modification of the learning design, which would improve the interactive personalisation of the learning process. This issue deals with how IMS-LD is interpreted by tools and engines developers and not with how the specification is actually designed.

8.8 . Authoring

As aforementioned, this thesis is focused on the specification itself and it does not deal with tools. However, authoring tools largely influence what can be modelled and how. Therefore, we point out a couple of key issues that could support the actual adoption of IMS-LD by the target groups:

a) There is a need for high-level visual authoring tools. Nowadays there are two types of tools: effective but too technical, even for technical profiles; and simple to understand but not powerful, since they usually deal with the very basic Level A. The creation of UoLs should be as far as possible from technical requirements or the underlying elements, components or structure. A more visual approach would encourage the understanding and use of IMS-LD in a broader sense by target groups. Technical low-level editors should live along with the visual high-level ones, though

b) Any authoring tool should allow for an integrated modeling, working with the manifest, the resources and the required external XHTML files with a common interface. It should dependencies and ease setting of properties. This is a hot challenge, not possible so far.

8.9 . **Solution**

We develop a technical set of solutions based on the aforementioned analysis and conclusions. These solutions deal with extensions, modifications and addition of modeling structures, elements and components, and the architecture of IMS-LD. More precisely:

- We develop a set of solutions focused on general-purpose modeling. These elements will be used as part of others specifically used in learning processes, like personalisation. Furthermore, they become a basic set to be re-purposed in different contexts and goals
- We develop a set of solutions focused on adaptive learning. A few very specific processes cannot be approached with just general structures. They need on-purpose elements which come across on-purpose goals on personalisation
- We develop a set of solutions that facilitate and improve the integration of Units of Learning and a bi-directional communication with other external resources, systems and standards. When needed, we highlight the need for a way of communication (e.g., a communication layer) although its development is something outside of the scope of this thesis. We are focused on the specification itself and how to improve the pedagogical expressiveness, and not on building any *ad hoc* technical artefact to get this aim through

Furthermore, we concentrate on solutions leaning on two categories: a) Modeling, that compiles every single extension, modification or addition, general or specific, to the specification and the information model; and b) Architecture, that deals with functional requirements of the spec, with a focus on the interoperability, communication and integration of IMS-LD with other external means. In both cases, we look for the highest performance along with the minimal structural change. Furthermore, we respect the original specification as much as possible and try to make as few changes as possible; on the other side, they all are needed to build the suggested solution, and cope with the main thesis.

Following, we show a summary of the review and the developed solutions (Table 21). Both can be found in detail in former sections.

Table 21. Developed solutions

ID	Issues, review and developed solutions
[M.05] [M.10] [M.11]	Programming structures and resources are very basic (simple condition, simple arithmetic, properties set-up, visibility, DIV layers)
[Ext.01a]	Condition type case
[Ext.01b]	Condition type case with automatic ranges
[Ext.02]	Conditional loop, type while
[Ext.03]	Integer loop, type for-next
[Ext.04]	Modification of the element <calculate>
[M.06]	There is no management of absolute time. There is no synchronization nor input point to work with relative time from
[Ext.05]	Modification of reference to relative time. Addition of reference to absolute time
[M.07]	Notification service, in Level C, is under-used. It only sends an email or plays an activity
[Ext.06]	Extension of the notification service, beyond using sendmail and playing an activity. It can be called from other structures besides the <on-completion> part of a learning activity
[M.08]	There is a blur way to handle the definition and use of properties and links amongst the several XML with global elements
[Ext.07]	Syntax modification, definition and use of elements view-property and

	set-property, as long as the properties which make use of them

[M.09]	Relationship between DIV layers and the visibility property is difficult to make and follow
[Ext.08]	In principie, the visibility property of any layer is turn off (hide), making simpler the condicional structure which could make use of it

[M.12]	There is no chance for iterations in any of the basic structures of the IMS-LD metaphor (learning activity, support activity, activity structure, act, play)
[Ext.09]	Extension of the current syntax of every element with a parameter <iteration> which defines a integer loop (type for-next) and-or a conditional loop (type while)

[M.13]	There is no synchronization input point in the manifest
[Ext.10]	Addition of an element GOTO which allows for a direct guiding of the learning flow

[M.14]	There is no chance to assign a specific activity to a selected user
[Ext.11]	Addition of an element ASSIGN-ACTIVITY-TO-USER which allows for a direct match amongst users, groups and roles, with learning activities and activity structures
[Ext.12]	Addition of an element ASSIGN-USER-TO-ACTIVITY which allows for a direct match amongst users, groups and roles, with learning activities and activity structures
[Ext.13]	Addition of an element SWITCH-ACTIVITY which allows for turning on-off activities and activity structures

[M.15]	There is no chance to make groups out of a selection inside the instance
[Ext.14]	Addition of an element CREATE-GROUP which allows for grouping users of the same role

[M.17]	The monitoring service does not allow for monitoring of groups
[Ext.15]	Extension of the monitoring service to trace roles and groups

[A.02]	IMS-LD does not allow for saving or retrieving data in external files, of any kind of format. In addition, connections with external databases or modules developed with other languages are not described or supported within the specification
[Ext.16] [Ext.17] [Ext.18]	Addition of the elements EXPORT and IMPORT to handle files with specific parameters (e.g., type TXT) and which is defined in a new property type FILE-IO
[Ext.19] [Ext.20] [Ext.21]	Addition of an the elements FROM-DB and TO-DB which allows for saving and retrieving data in a database of type MySQL. The connection is defined in a new property type DATABASE

[A.04]	There is no chance to modify the learning skeleton, method, roles definition or any other structural element in run-time
[Ext.22] [Ext.23] [Ext.24] [Ext.25]	Addition of two couples of global elements: a) *view-IMS-LD* y *set-IMS-LD*, b) *view-resources* y *set-resources*, which allows for the visualisation and modification of the learning design and the related resources in run-time

La sabiduría no reside en comprender cosas

sino en comprenderse a sí mismo

Han Fei-tzu, Tao Tê Ching (S.VI, ac)

9. Annex I: other aspects related to adaptation, integration and IMS-LD

9.1 . Visual interfaces

This section is focused on Using the IMS Learning Design notation for the modeling and delivery of education and it is co-authored by Colin Tattersall and Tim Sodhi (Open University of The Netherlands) [243]

IMS Learning Design (IMS-LD) is a notation system for learning and instruction. It supports the description of learning processes using a set of standardised concepts, including roles, activities, acts, objectives and prerequisites. With the availability of such a notation, descriptions of learning processes can be shared, critiqued, modified, rated, compared and evaluated. Moreover, the machine-interpretable nature of the notation means that designs can be executed by software to support the dynamic orchestration of multi-learner, multi-role learning processes. This chapter introduces IMS-LD and describes experience with its use, supported by the first generation of tooling. We then combine these experiences with observations on the tools in the light of new developments in e-learning in order to derive a set of requirements for IMS-LD-enabled visual design environments.

9.1.1. Introduction

In a recent paper, Merrill [244] highlights that training is often created by designers-by-assignment without the use of a systematic process, and that instructional products are often designed without sufficient consideration of the applicable instructional design theory. Other research indicates that even when designers are aware of theories, there appears to be a difference between their practice and instructional design models [115; 245; 246]. Part of this problem is the absence of a tradition of the use of notations [247-249]. In order to address this issue, several initiatives have been pursued to derive a modeling language for education [34; 55; 250]. The results of these initiatives, notations for describing educational processes, have been input to standardisation processes [251] and, in 2003, an open technical specification known as IMS Learning Design [81], was approved by a consortium of universities, system vendors, providers and other e-learning stakeholders.

In Waters and Gibbons' [247] terms, IMS-LD can be positioned as a notation system. The notation is characterised in [25] as a layered, formal, textual specification offering a single perspective. This chapter describes the IMS-LD notation system and reviews experience with its use. We then identify a number of requirements for IMS-LD-aware design environments with broad utility, a high degree of usability and support for interoperability.

9.1.2. IMS-LD: a notation system for education

IMS-LD focuses on the creation of a formal description of educational processes known as a Unit of Learning (UoL). In practice, UoLs define the set of learning activities, for example courses, assessments, workshops or seminars in a specific pedagogical setting and can serve various functions depending on the learning objective and design [252; 253]. A wide variety of pedagogical approaches can be represented by IMS-LD, such as problem based learning, competence based learning and game based learning. Prior to turning to the details of the specification, it is helpful to review the requirements the specification was written to meet:

1. Completeness: describe the teaching-learning process in a UoL, including references to the digital and non-digital learning objects and services needed during the process.

2. Pedagogical Flexibility: describe different kinds of pedagogies without prescribing any specific pedagogical approach.

3. Personalisation: describe personalisation aspects within a learning design, so that the content and activities within a UoL can be adapted based on the preferences, portfolio, educational needs, and situational circumstances of users. In addition, the control over the adaptation process must be given, as desired, to the student, a staff member, the computer, and/or the designer.

4. Formalization: describe a learning design in the context of a UoL in a formal way, so that automatic processing is possible.

5. Reproducibility: describe the learning design abstracted in such a way that repeated execution in different settings with different persons is possible.

6. Interoperability: support interoperability of learning designs.

7. Compatibility: use available standards and specifications where possible.

8. Reusability: identify, isolate, de-contextualize and exchange useful learning artefacts, and to re-use these in other contexts.

The specification meets these requirements by defining a modeling language. The specification defines the various concepts in the IMS-LD language in an Information Model—descriptive text together with diagrams of the relationships between the concepts expressed in the Unified Modeling Language (UML) [254]. Advice and explanation on putting the language to use is included in the Best Practice and Implementation Guide, and a representation is given as an XML Binding—whereby the learning design and the concepts specified, are represented in machine interpretable XML documents.

The specification prescribes a standardised, flexible language for representing learning scenarios for multiple or individual learners, able to be executed by software responsible for coordinating learners, teachers, learning resources and activities as the learning process progresses. The specification reflects a general model which underlies many different behaviourist, cognitive, and (social) constructivist approaches to learning and instruction: people act in different roles in a teaching-learning process. In these roles, they work towards certain outcomes by performing learning and/or support activities within an environment, consisting of learning objects and services to be used during the performance of the activities. A learning object is defined by the IEEE LTSC (2000) as, "any entity, digital or non-digital, that can be used, reused or referenced to during technology-supported learning." The approach separates learning objects and services from the educational method used in the UoL. Put succinctly, IMS-LD allows designers to specify who should do what, when and with which support facilities in order to reach learning objectives.

Which concepts, then, does the specification prescribe? Central to the modeling language are the activities to be carried out by learners and staff (or other roles involved in the learning process). An activity is associated with learning objectives, prerequisites, a description and an environment. Environments include the material, tools and facilities needed by learners and staff in order to carry out their activities, and include learning objects (such as documents, explanatory videos, and animations) and learning services (such as forums and chat facilities). These core components (roles, activities and environments) are orchestrated in a method. Methods use the metaphor of a theatrical play to describe the temporal flow of events, whereby sequential acts are described, with the play ending with the completion of the last act. The transition from one act to another serves as a synchronisation point for the multiple participants, ensuring that they can all start the next act at the same time.

In addition to the basic language constructs, referred to as level A, the specification provides additional concepts to cater for more sophisticated process descriptions. IMS-LD levels B and C [61] allow the expression of conditions, so that designers may describe circumstances under which specified actions should follow. For example, a designer may wish to arrange for the learning process to accommodate different paths through learning activities depending on the results of a self-assessment by learners, or given learners' preferred approach to learning. Additional constructs allow information to be stored during a learning process and used subsequently to influence the flow of events. In this way, designers can arrange for peer review of documents, whereby one learner's contribution is rated and critiqued by peers before being returned to the original author for reflection. The ability to notify roles that an event has happened or that intervention is required is also afforded by the IMS-LD modeling language.

The language constructs offered by IMS-LD allow a wide range of educational processes to be modelled in a standardised way [255]. Use of the language provides the basis to rejuvenate e-learning systems, increasing the 'richness' of learning activities. New, more effective, efficient and attractive learning models can be specified (e.g. active learning, problem based learning) giving specific attention to support of the teaching/learning process to decrease workload (particularly that of teachers). Furthermore, the advantages of a standard notation can be realised: reflection, communication, sharing, reuse, research, similarity studies and evaluation.

These benefits are, however, predicated on more factors than those addressed by the open technical specification itself. Just as the HTML specification [256] describes the constructs offered by the HTML language without specifying the nature of the software which interprets it, the IMS-LD specification does not address how to record or create the notation, how to adapt or edit it, how to aggregate several use of the notation and other factors involved when putting IMS-LD to use.

9.1.3. **IMS-LD design process and tools**

As noted in the abstract, e-learning production processes [257] have sub-processes in which Units of Learning (UoL) are developed and stored, populated with specific learners and teaching staff, and, to continue the theatrical analogy, performed, or 'run'. Since UoLs do not relate to specific individual learners and teaching staff, they can be created once and delivered many times [248].

The IMS-LD specification prescribes the form and structure of UoLs so that software applications may be created for their interpretation. As noted above, XML is used as the machine interpretable language in which UoLs must be described to be IMS-LD compliant. A fragment of the XML representation of a UoL is shown following:

```
<play identifier="P-1" isvisible="true">
  <title>A Unit of Learning on the European Constitution</title>
  <act identifier="A-1">
      <title>Views on the European Constitution</title>
      <role-part identifier="RP-Learner-1">
        <title>Learner RP</title>
        <role-ref ref="Learner"/>
        <activity-structure-ref ref="AS-first-step"/>
      </role-part>
      <role-part identifier="RP-Facilitator-1">
```

```
        <title>Facilitator RP</title>
        <role-ref ref="Facilitator"/>
        <support-activity-ref ref="SA-first-step"/>
      </role-part>
    <complete-act>
        <when-role-part-completed ref="RP-Facilitator-1"/>
    </complete-act>
  </act>
  <complete-play>
      <when-last-act-completed/>
  </complete-play>
</play>
```

The software which interprets this XML notation is referred to as an IMS-LD engine [258; 259], a software service used by an IMS-LD player [40], the interface with which learners and staff interact. The engine makes the appropriate activities and environments available to people playing the various roles, through the player, coordinating and synchronizing the dynamics of a learning process as multiple learners work through a learning process. The distinction between engine and player allows a variety of approaches to the look and feel of interaction in learning processes (different players) to be supported by a single orchestration service (engine).

Figure 62 shows a member of staff interacting with an IMS-LD player.

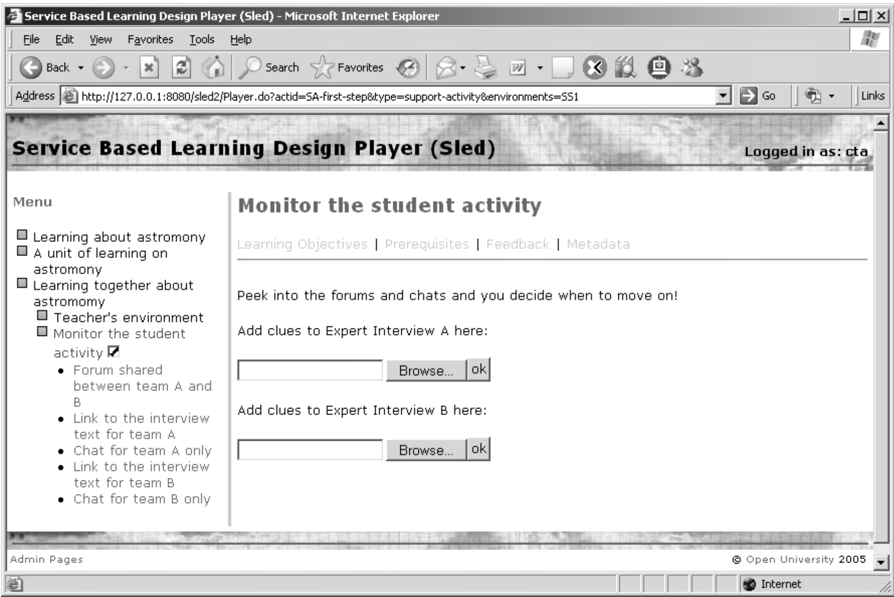


Figure 62. An IMS-LD player interpreting a UoL

The player shown in Figure 62 is a web server-based application. Once the user has surfed to the appropriate web address and been authenticated, learning or support activities are made available. Note that what is seen by a particular user (learner or staff member), is a personalised, specific set of activities and associated environments according to the design recorded in the IMS-LD notation and the current state-of-play of the learning process. The content of the web page is generated from the ongoing processing of the notation by the engine. Drop-down boxes, forms and buttons are all derived from concepts in the language. Whether and when the learner is presented with a given activity depends on the conditions specified by the designer of the learning process.

So, along which processes and using which tools do designers create UoLs? The IMS-LD Best Practice and Implementation Guide offers a suggestion for a three-step approach to the creation of UoLs. Individuals or teams involved in the design of e-learning start with a narrative description of the proposed learning process. Unified Modeling Language Activity Diagrams are used to make the flow of events within and between roles explicit, before a final phase in which the XML-based UoL is created. Sloep, Hummel and Manderveld [260] offer a related procedure for UoL design, and Janssen and Hermans [261] provide experience with the approach in a distance learning context.

Applying such an approach to designing within a Problem-Based Learning context, a narrative description would be:

- The coordinator for the course makes a problem description available to the group.
- Each of the students in the group reads the problem, as does the facilitator.
- The students decide on a chairperson, the spokesperson for the group, responsible for recording key group decisions, and the chosen representative is appointed as such by the facilitator.
- The group then communicate amongst themselves to clarify the problem, using each other and the facilitator to discuss and clarify terminology and any open issues, eventually arriving at their own succinct statement of the problem at hand.
- The chairperson states this problem description in a document and the group continues by identifying possible solutions or explanations for the problem.
- These possible explanations are clustered into a small number to be explored further by the students.
- The explanations to be pursued are listed in a document.
- The group then identifies the learning goals of the problem and individuals embark on the required research.
- Eventually, the group meet up to discuss their findings, again assisted by the facilitator.

• The chairperson summarizes the findings in a document.

• Subsequently, an Evaluator and the Facilitator discuss the performance of the group and the Evaluator provides an Evaluation of the group.

A transformation of the above narrative to an activity diagram is shown in Figure 63.

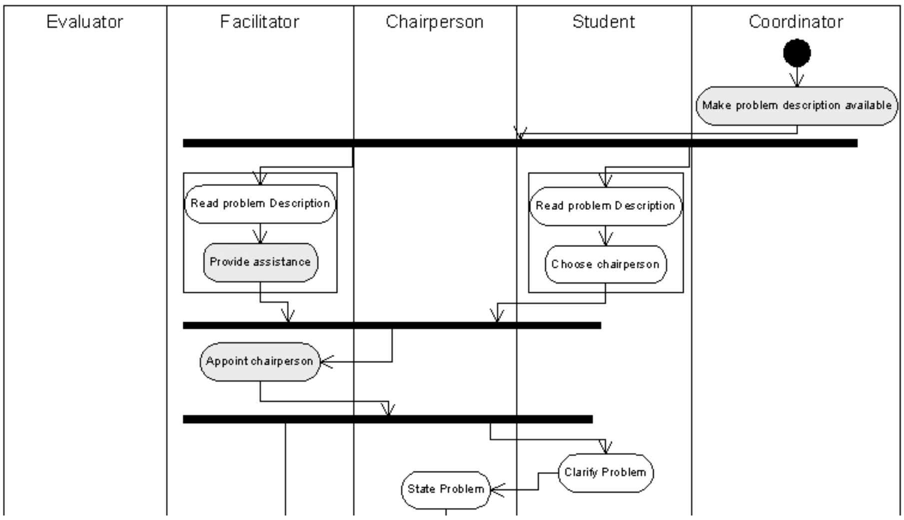


Figure 63. A partial activity diagram for Problem Based Learning

Turning to tooling, the XML illustrated before can be created with text-based editors such as Windows Notepad. This is, of course, a time-consuming and error-prone activity. Syntax problems (for example, missing angled brackets and string quotes), the need for authors to maintain a mental list of XML element names and long “create-test-debug” cycles reminiscent of the early days of computer programming languages, all indicate the need for a more supportive environment for designers. The use of XML editors alleviates the difficulties to only a small degree. Template-based approaches such as that described by Janssen and Hermans [261] either require the development of significant tooling skills by learning designers or a team-based approach which includes notational specialists.

A number of initiatives have been carried out to create a higher degree of support for the creation of UoLs. Reload [69] is a tree and form-based editor which has seen significant use in the IMS-LD community. Figure 64 shows the editor in use.

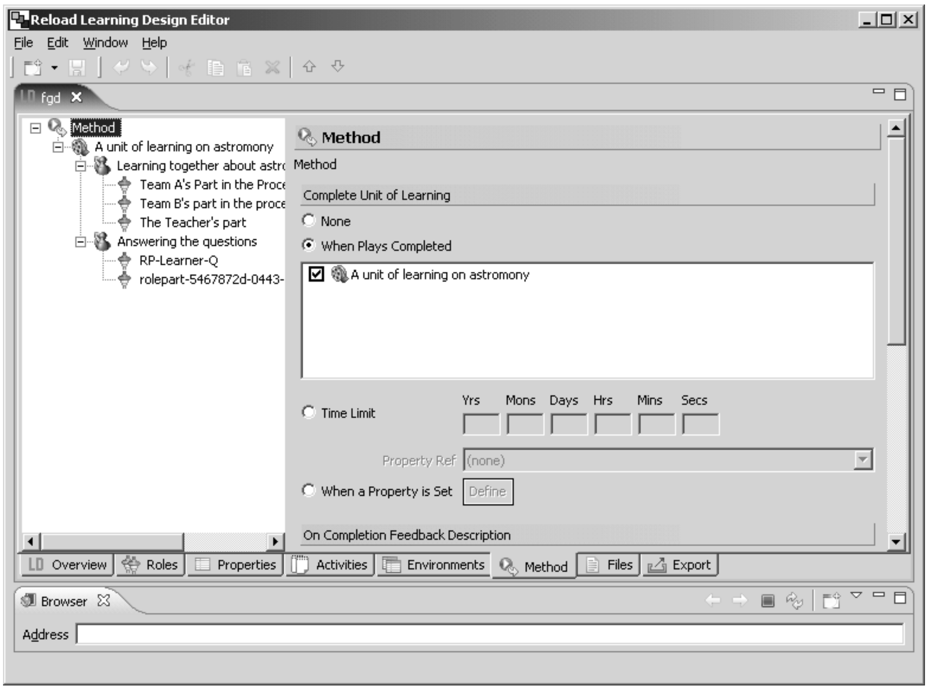


Figure 64. The Reload LD Editor

Reload takes the learning designer away from the intricacies of the XML binding, organising the user interface around core IMS-LD concepts (the tabs for overview, roles, properties and so on) using forms to gather data through particular controls (check boxes, drop down lists etc). Designers' work can be checked for completeness with respect to the requirements on a UoL, and can be exported as a UoL content package. Once in this format, the UoL can be uploaded to an IMS-LD player environment and tested.

Experience with using Reload [262] highlights a steep learning curve and a need to resort to combinations of editors (e.g., Reload plus Notepad) to cope with IMS-LD level B. Fundamentally, since Reload was written as reference implementation of an IMS-LD editor, everything which is possible to be encoded in IMS-LD notation (whether using Reload or another IMS-LD editor) must be able to be imported, adapted and exported. This leads the tool to be close to the specification [38], requiring the user to be familiar with the notation. A variation on this style of support can be seen in the CoSMoS editor [263], shown in Figure 65, and in CopperAuthor [50], shown in Figure 66.

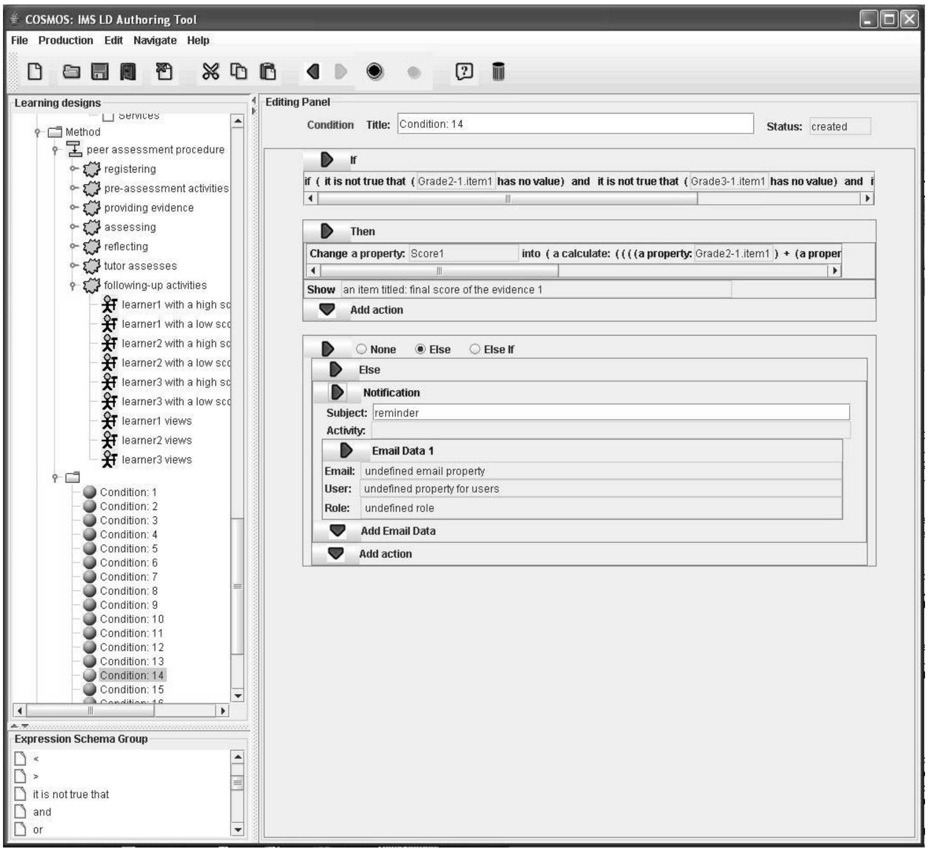


Figure 65. The CoSMos Editor

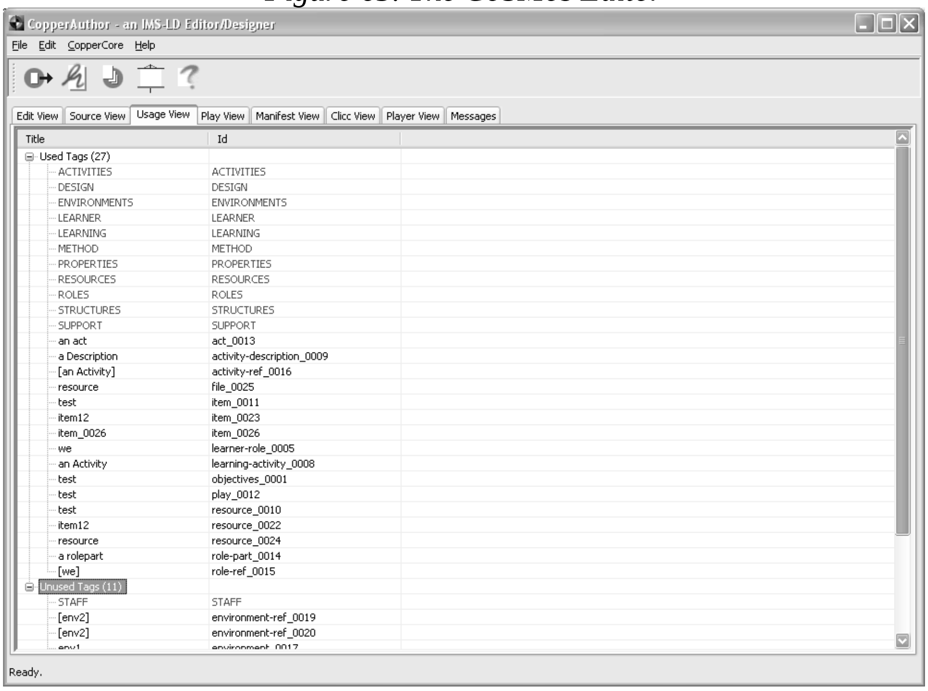


Figure 66. CopperAuthor

Yu, Zhang and Chen [264] highlight the need to support those who may have little or no knowledge of the IMS-LD notation yet are involved in the design of e-learning, giving teachers as an example target group. The authors cite MOT+ [33] and ASK-LDT [265] as more appropriate for this user group. Figure 67 and Figure 68 show the graphical approaches used by MOT+ and ASK-LDT.

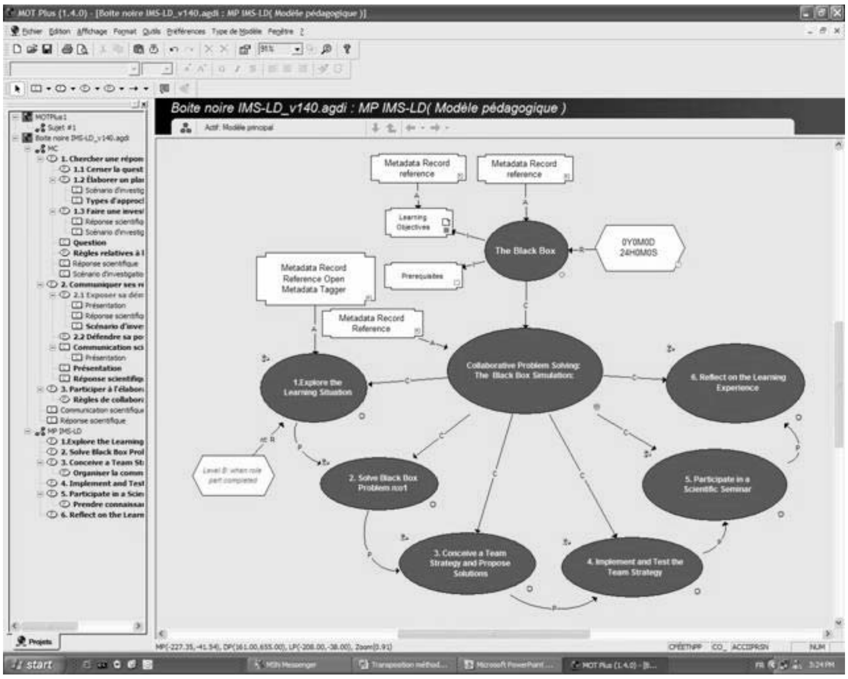


Figure 67. The MOT+ environment

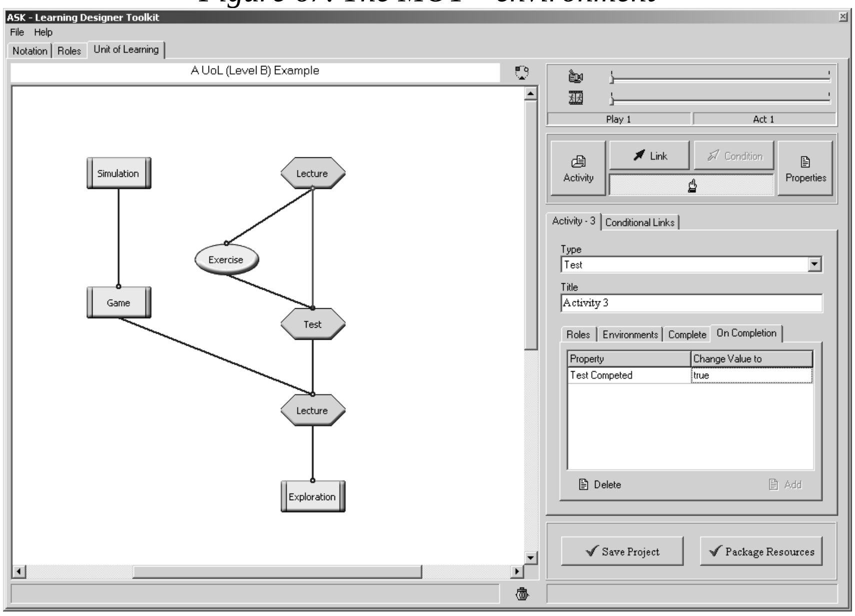


Figure 68. The ASK-LDT environment

Although certainly offering a more protective environment in which designers can work, caution is needed before concluding that use of graphical interfaces will remove barriers to UoL production. Spang, Bovey and Dunand [266] highlight that modeling a complex sequence of activities with MOT+ is "too disconnected from the daily practice of the average teacher." The issue of the part played by teachers' in production of UoLs is explored in some detail by Griffiths and Blat [267]. The authors share the opinion of

Spang, Bovey and Dunand that, while the underlying concepts of the IMS-LD modeling language are not complex, they differ from the concepts that a teacher uses to plan educational activities. Two challenges are identified in the article: supporting teachers during preparatory stages and helping teachers author and edit UoLs. A number of approaches to meeting these challenges are explored by the authors including the use of patterns (more on this in the following section), templates, primitives, and placing constraints on the available design options.

Putting IMS-LD to use today, then, is done along multi-step UoL production processes supported by a number of different editing environments, engines and players.

9.1.4. **Some observations**

A number of observations can be made, reflecting on the various initiatives in the area of IMS-LD-aware tooling, as well as the learning design process.

9.1.4.a. ***IMS-LD design process***

As mentioned in a previous section, the IMS-LD Best Practice and Implementation guide offers suggestions for a three-step approach to creation of UoLs. The approach was illustrated using an example from the world of Problem Based learning. The question how does a designer creates quality and effective UoLs using design rules that can be used to capture knowledge and assist in developing the best suited learning design, bodes further study. A learning design rule describes the learning method that can be applied to a specific learning situation with a certain probability of success [268]. Three approaches have been specified to capture learning design rules: (1) using designs based on instructional design theory; (2) using designs based on best practices in instructional design; and (3) using designs based on patterns in best practices.

In the first approach, the designer is cognizant of the Instructional Design approach that would appropriate for the specified learning objective. An example of this approach was described earlier in this chapter, with regard to Problem Based learning. The designer can create the course based on prescriptions in the Instructional Design approach. While this approach works best for Instructional Designers with a clear and explicit knowledge of Instructional Design theories and models, the same is not necessarily put centre-stage in the second approach, where the designer looks for comparable successful example UoLs, and can customize the same based on the current learning objectives. The idea of reusable learning design, led to a study conducted by Bennett, Lockyer, & Agostinho [269], which demonstrated the efficacy of adapting successful learning designs, to suit various learning contexts, in the creation of quality courses.

There is at present, substantial interest in the application of pedagogical design patterns to learning design. Here, the designer searches for course patterns which represent “best practice” in terms of both pedagogy and use of Information and Communication Technologies [269; 270]. Patterns can then be chained together like building blocks to create a new course. Pedagogical patterns capture effective learning designs, allowing newcomers and non-experts in education to learn from more experienced developers, providing solutions that can be applied in many learning scenarios, functioning as a communication medium and means and enhance knowledge management and knowledge transfer [268; 271-274]. It has been argued that the point of patterns is not to support immediate reuse, but rather to support creativity, supporting the practitioner engagement rather than relieving practitioners of pedagogic responsibility. This view finds resonance with Olivier [275], who expressed that the ability to share and modify learning designs will enable the building of better practice for e-learning. The Collage system [276] builds upon this approach to model Collaborative Learning in IMS-LD.

9.1.4.b. ***IMS-LD tool support***

The first generation of IMS-LD-aware tool supports only part of the design process. The early stages, in which ideas are sketched, initial narratives developed and storyboards worked out, tend to fall outside the tools’ scope. Similarly, the tools tend to offer partial support for the development of UoLs, relying on separate tools for the creation of textual and graphical content, assessments, simulations etc. The design process using IMS-LD today, involves writing down a design as described in the best practice guide, and involving building use cases, representing activities through UML diagrams (suggested, but not compulsory) and then codifying the design in XML—which is at best time consuming [277; 278]. Only the last stage of this is apparent in the current range of software tools. While pragmatic in terms of reusing existing tools and other initiatives, this approach tends to lead to a rather fragmented design experience and limited practitioner engagement. Once created, these components must be integrated into an orchestrated learning process described by the learning design of a UoL. Moreover, the issue is not only one of linking to passive resources on the web, but integrating and orchestrating active components into a learning process (e.g., posts to a blog, new entries in a wiki). What happens in an assessment, or in a simulation, should be able to be used to influence the rest of the learning flow.

Most of the first generation tools discussed, are geared towards the IMS-LD community, and serve more as reference implementations of the specification (e.g. Reload, CoSMoS, etc.). An author working with these tools needs to possess detailed knowledge of the elements of IMS-LD and their function in order to create a UoL. The next generation of IMS-LD tools endeavour to hide the specification from the end-user, allowing the user to realize the learning design process without necessarily being exposed to the underlying complexity [279]. The endeavour in the IMS-LD community currently is on developing tools that, being standards compliant in their outputs, are not standards oriented in the interface they present to the end-user (Griffiths & Blat, [280].

Although most of the tools have seen evaluation, these have often been small-scale studies, not investigating the practitioner acceptance and usage of the tools. A body of evidence on how designers use today's tools is still being accumulated. But who are the users of these tools? Some tools explicitly address technically-savvy users who are interested in using the tools to learn more about the specification rather than for use in educational production processes (e.g., Reload). Others state the target group as being teachers. This group is, however, extremely diverse and may rely on very different production processes and need a variety of tools. Teachers in primary education have different approaches to e-learning than those in vocational education, which differ again from those in higher education. Even within a particular type of education, different approaches are seen, such as the faculty-driven model and design-team driven model in higher education, described in [281]. While the use of the word teacher suggests a traditional academic setting, e-learning is heavily applied in corporate training and human resource development environments [282; 283]. Designers in these environments may well require different support environments. Moreover, the group "designers of e-learning" is broadening as those outside traditional educational and training contexts use networking technologies help others learn. This help takes the form both of suggestions or recommendations of how to approach the attainment of a particular competence, as well as the full "recreational design" of learning processes (such as designs for learning languages, sports, musical instruments, project management, plumbing, architecture and so on).

Part of the challenge when designing e-learning (whether by schooled instructional designers, designers-by-assignment or recreational designers) revolves around how to take advantage of theoretical insights and best practices. Merrill and Wilson argue for embedding principles of effective and efficient learning design so that tools "provide intellectual leverage to designers who may not know the required instructional design theory" [244]. Although such principles can be encoded in IMS-LD, exposing designers to the raw notation does not give the required leverage. In this context, Waters and Gibbons [247] refer to the intuitive versus non-intuitive dimension of notation systems, and the trade-off which needs to be stuck between human and computer use of a notation.

Could IMS-LD, then, be married to a standardised graphical notation which would appeal to the intuition of designers, allowing them to express the learning design in concepts familiar to their design practices? Which could then be mapped to IMS-LD concepts?. This abstraction is analogous to creation of HTML pages in WYSIWYG HTML editing tools like Microsoft FrontPage, Adobe Dreamweaver, etc. Here the user may have limited to no knowledge of HTML code, but can create rich webpages quickly. This should not however be interpreted as a mere representation of IMS-LD constructs in a graphical interface. These concepts or representations used can be aggregations of activities and roles which are then mapped to IMS-LD constructs (Es & Koper, 2006; Griffiths et al., 2005). Mappings could then be defined between the graphical notation and IMS-LD allowing translations between the two, and visual design environments could be created to support the notation. The comparison to WYSIWYG HTML editors, it should be noted, is not rigorous because different WYSIWYG HTML editors do share common vocabulary and concepts like pages, fonts, tables etc.—a common notion that is, however, missing in today's implementations of IMS-LD editors and design environments.

It seems however, that faced with a large diversity of users, a single intuitive visual notation may not suffice. Experience with UML [284] suggests its models are no panacea for the many and varied stakeholders involved in system development. We note, however, that diversity of graphical notations, as is the situation today with ASK-LDT and MOT+, makes the interoperability situation more complex—today's tools all export IMS-LD able to be interpreted by compliant engines, yet round-tripping between the tools is not possible. Supporting variety in visual notations while preserving interoperability is a key research topic for the next generation of visual design environments.

9.1.5. Conclusion: requirements for a new generation of interoperable e-learning design environments

With these observations in mind, we list with a series of requirements that form the basis for our ongoing research in the area of visual design environments for IMS-LD:

- There is a need for end-to-end support of design processes from idea formation through to complete UoL (see Botturi, [30] for results in this area).
- IMS-LD Design Environments should support holistic e-learning design, incorporating formative assessment, simulations, multimedia content and other parts of an e-learning experience. This is not merely a question of linking to objects or services on the web, but requires designers to be a position to specify which information should be passed to, and taken back from, learners' interactions with content and services.
- E-learning designers should be shielded from the intricacies of notational bindings. Wizards, templates, alternatives metaphors and techniques from the world of visual design environments, can all help in meeting this need. Moreover, research in End

User Development [285] offers a number of pointers to address this problem. A closer involvement of groups of designers in the development of environments (rather than solely in their evaluation) would seem appropriate.

- Design Environments should accommodate a high degree of variation in designers' knowledge and experience with pedagogies, both traditional and those focused on e-learning. Although some groups of designers may require extensive handholding, Hoogveld [286] notes that teachers do not like prescriptive methods. Having the flexibility for designers to experiment with, tune and indeed, create templates, patterns and primitives might help strike the correct balance between too restrictive an environment and an unsupportive one.
- Finally, we emphasise the need for design environments to be created to be interoperable. Without the capability to both import and export standardised notations such as IMS-LD, tools users become shackled to a particular design tool, fragmenting the community and creating competition where cooperation would offer more benefits.
- Creating environments to meet these needs will help reach the goals of sharing, critiquing, modifying, executing, rating, comparing and evaluating learning designs across the broad spectrum of e-learning designers and teachers.

9.2 . Templates and learning scenarios

Esta sección se centra en el modelado y uso de plantillas y escenarios de aprendizaje en entornos b-learning desde la práctica educativa y está co-escrita con Gemma Corbalán Pérez (Open University of The Netherlands) [62].

El mercado actual del aprendizaje online marca una separación entre la teoría educativa y la práctica educativa real. Los escenarios de aprendizaje diseñados y utilizados por los profesores tratan de congeniar la enseñanza presencial con las tecnologías de información y comunicación creando nuevos escenarios mixtos que no son contemplados del todo en la teoría educativa existente. En una primera parte, este artículo ahonda en la literatura sobre teorías educativas, define qué se entiende por escenario de aprendizaje y por aprendizaje mixto o blended learning y resalta la separación entre la teoría y práctica educativa en contextos educativos reales y las consecuencias asociadas. A continuación, se describe el estudio práctico llevado a cabo con dos grupos de profesores universitarios que utilizan escenarios mixtos y los patrones de enseñanza obtenidos. Para finalizar, se aporta una solución concreta al modelado de este tipo de escenarios mediante la integración de la especificación sobre eLearning IMS Learning Design, el sistema gestor de cursos Moodle y la visualizador de Unidades de Aprendizaje Sled.

9.2.1. Introducción

Consideramos un escenario de aprendizaje como un conjunto de actividades, recursos y métodos que refleja una unidad de aprendizaje o lección [85] [227]. Adicionalmente, un escenario de aprendizaje puede representar un caso que simula situaciones reales de una manera controlada [287] con el objeto de familiarizar al estudiante con conceptos, contenidos o procesos dentro de un contexto significativo y relacionado con su utilización profesional posterior [288]. En cualquier de los das acepciones, el escenario representa una situación de aprendizaje manifestada en una lección o curso, mediante la definición de roles, actividades, recursos y herramientas [289]. Estas situaciones pueden centrarse, por ejemplo, en la recopilación de información, la aplicación de una teoría o de una técnica, la realización de una tarea, el análisis de un suceso o la toma de decisiones [290]. Constituyen pues un instrumento de modelado y asimilación de situaciones o momentos de aprendizaje, pero nunca el objetivo último de la enseñanza [291].

En función de esta definición de escenarios de aprendizaje podemos afirmar que existe un número considerable de escenarios de aprendizaje basados en la experiencia de profesores que no se encuentran fundamentados en la teoría instruccional [292]. La mayoría de teorías tradicionales, como Constructivismo, Cognitivismo o Comportamiento, suelen ser las utilizadas para generar entornos teóricos instructivos [293]. Es decir, dichas teorías tienen como propósito la comprensión e identificación de los procesos mentales que hacen posible el aprendizaje y, a partir de ellos, tratar de describir métodos para que la instrucción sea más efectiva. Sin embargo, la realidad es que los escenarios de aprendizaje que se aplican realmente en la práctica no están fundamentados en dichas teorías educativas. Esta insuficiencia de fundamentación teórica conlleva una inconsistencia de base que resulta en la imposibilidad de replicar las experiencias para su posterior generalización [294]. Entre las razones existentes por las que parece existir estar separación entre teoría y práctica educativas destacamos: a) Las prescripciones teóricas no son siempre aplicables en la práctica y pueden ser demasiado teóricas para que los profesores reales puedan aplicarlas en la práctica [292] y b) las prescripciones teóricas pueden haberse quedado obsoletas en el contexto de rápido avance tecnológico en el que vivimos [295].

Por otro lado, la utilización de patrones de diseño de aprendizaje con requerimientos y métodos comunes pero adaptables para cada estudiante en función de su característica intrínseca en función de sus necesidades educativas e/o intereses forma uno de los pilares de la moderna teoría instructiva sobre adaptación [296; 297]. Esto es, partiendo de un modelo genérico que cumple con un marco común, este se aplica individualmente o por grupos de estudiantes con similares necesidades educativas e/o intereses. Pero esta generalización de escenarios de aprendizaje diseñados desde la práctica no es posible si tomamos de manera separada las herramientas de creación y uso de las teorías instruccionales. Es necesario un matrimonio entre práctica y teoría que permita la fundamentación de la realidad didáctica para fomentar una posterior retroalimentación y una extensión de dicha teoría así como una generalización de la práctica en sí misma.

A este respecto, y siempre desde la perspectiva del aprendizaje online (total y parcial -presencia-online-, también llamado mixto o blended), diseñar escenarios pedagógicos reales fundamentados en una teoría instruccional sólida nos permitirá obtener posteriormente una conceptualización de los mismos, abrimos el camino a la definición de patrones de enseñanza y de aprendizaje con las consiguientes ventajas, como son la reutilización y la interoperabilidad. Reutilización de un patrón de enseñanza en cuantas situaciones de aprendizaje se desee, con la consiguiente contextualización a la realidad concreta del alumno, del grupo de alumnos, del profesor o del centro de enseñanza. Interoperabilidad de un patrón de enseñanza generado para el aprendizaje online o para el aprendizaje mixto que permite la modificación y utilización del mismo escenario bajo diversos requisitos y configuraciones técnicas manteniendo la atención en los conceptos y contenidos y no en las limitaciones o particularidades tecnológicas [58].

9.2.2. **Caso práctico**

En diciembre de 2005 y marzo de 2006 se realizaron dos talleres en la Universidad Complutense de Madrid (co-dirigido por Campus Virtual de la Universidad Complutense de Madrid y la Open University de Holanda-OUNL) y en la Universidad de Cádiz (co-dirigido por su Facultad de Ciencias de la Educación y OUNL) centrados en la definición y modelado de escenarios de aprendizaje desde la práctica docente. En los talleres participaron un total de setenta profesores universitarios. Durante las sesiones se trabajaba sobre las necesidades de los escenarios mixtos, en el modelado de Unidades de Aprendizaje que permitieran incorporar las tecnologías online a las clases presenciales, en la migración de sistemas propietarios de gestión de cursos (básicamente WebCT) a sistemas de código abierto u open source (Moodle) y en el análisis constructivo de características y propuestas de mejora. Cada participante, entre otras tareas, debía diseñar al menos un escenario de aprendizaje que reflejara una unidad de aprendizaje que se ejecutara en su realidad particular docente y en donde incorporara tecnologías online. La unidad de aprendizaje podía abarcar cualquier entidad significativa dentro de una asignatura (un tema, un cuatrimestre, la asignatura completa...). Posteriormente, se realizó un análisis de todos los escenarios entregados con el objeto de identificar los patrones de enseñanza mixta empleados y se procedía al modelado y ejecución de los mismos con IMS Learning Design (una especificación pedagógicamente flexible con la que el diseñador de aprendizaje puede representar un escenario basado en cualquier teoría instruccional, como explicaremos en el siguiente apartado).

Se obtuvieron un total de ochenta y cinco escenarios, disponibles en [298; 299]. Todos ellos se pueden categorizar en cuatro patrones de enseñanza mixta utilizados por la muestra (Figure 69):

- Escenario de aprendizaje 1. Donde los recursos tecnológicos son utilizados en paralelo a las actividades prácticas
- Escenario de aprendizaje 2. Donde los recursos tecnológicos son utilizados además durante los momentos de conclusión y evaluación
- Escenario de aprendizaje 3. Donde los recursos tecnológicos son utilizados en paralelo durante toda una unidad de aprendizaje centrada en el debate
- Escenario de aprendizaje 4. Donde los recursos tecnológicos son utilizados en paralelo a todas las actividades presenciales centradas en un trabajo de investigación

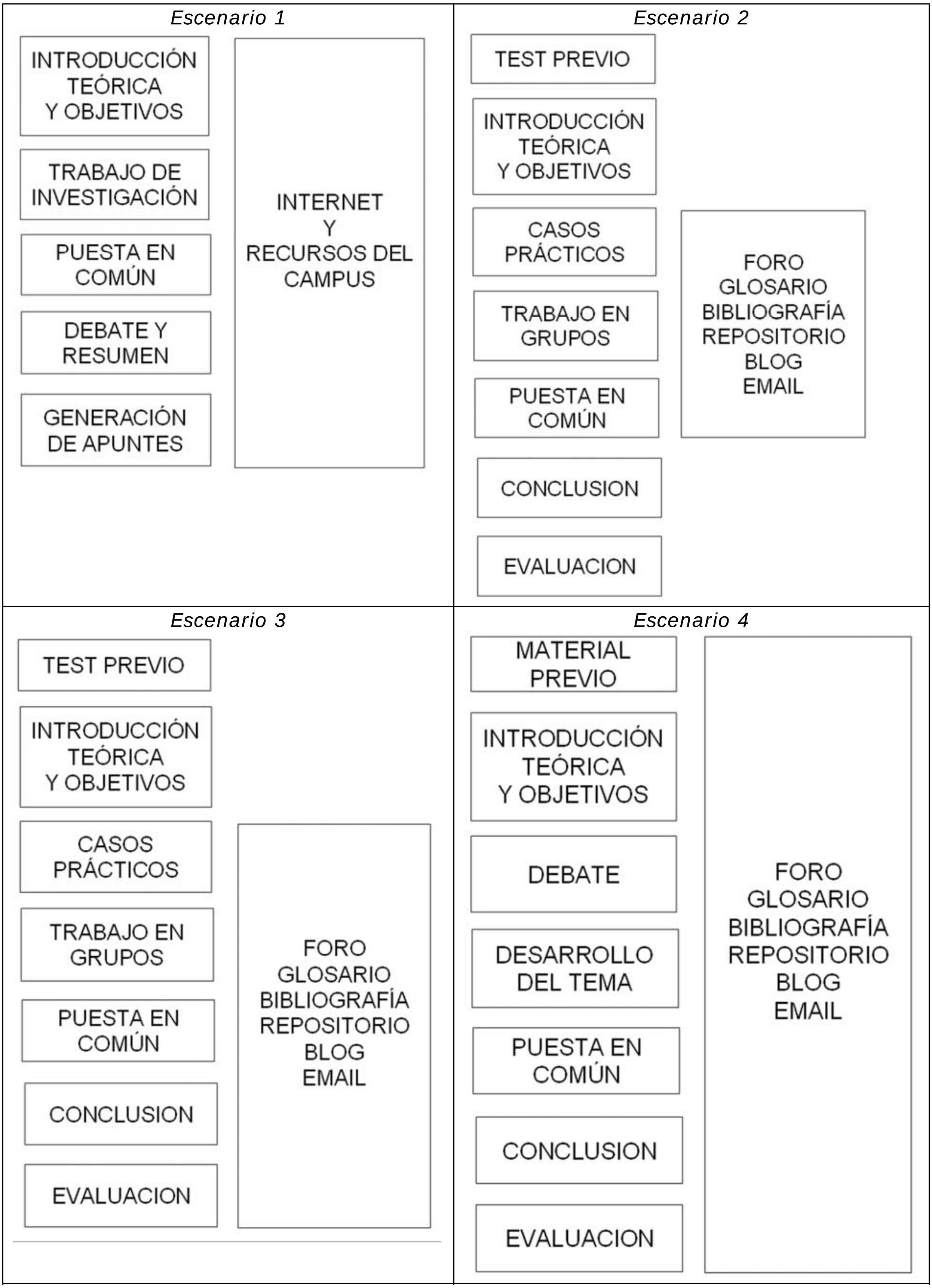


Figure 69. Escenarios de aprendizaje 1 a 4

Asimismo, las actividades presenciales se nutren de debates, introducciones, puestas en común, casos prácticos, trabajos en grupo, tests, evaluaciones y conclusiones y los recursos tecnológicos se nutren de herramientas online, como foro, glosario, bibliografía, repositorio, blog o email.

La lectura obtenida de los escenarios proporcionados, así como de los cuatro patrones generales, aporta los siguientes puntos fuertes:

- Existe una identificación de actividades y distinción de funciones entre profesor y alumno. Los roles se encuentran perfectamente marcados
- Existe una identificación clara de momentos educativos y actividades
- Para clases de una sesión, existe una mezcla de teoría-práctica, de explicación-debate, pero con utilización anexa de la tecnología en momento separado
- Para clases de más de una sesión, la secuencialidad viene marcada por el calendario
- Existe trabajo con anticipación: Leer y prepararse antes de clase. El recurso online se utiliza para centrar al estudiante
- Existe trabajo con refuerzo posterior: Aportar conclusiones y seguir debate en foro. El recurso online se utiliza para reforzar conceptos y mantener activo el grupo y el tema
- Existe trabajo en paralelo presencia – online, aunque no haya una conexión real entre ellos. El recurso online es un elemento añadido y no forma parte del proceso troncal del itinerario de aprendizaje

Del mismo modo, los datos obtenidos también perrmiten entrever algunos puntos poco presentes o ausentes del todo :

- Poca variedad en el uso de herramientas online. Mayormente, repositorio, foro y correo, a veces blog. Otras herramientas como wiki, chat online, multiconferencia, autotest, juegos o simulaciones son del todo ausentes
- Desarrollo de varias actividades en paralelo, online y presencial. Se busca la secuencialidad en el flujo formativo. No se potencia las actividades en paralelo
- Trabajo online en grupo. Se centra en modelos individuales sin potenciar las herramientas colaborativas
- Trabajo inductivo. Se utilizan los recursos online como un complemento, pero generalmente no como una fuente que aporte a la presencia y al discurso pedagógico
- Tarea investigadora del estudiante. Salvo en contadas ocasiones (Escenario 4) no se promueve la investigación online por parte del estudiante

- Evaluación formativa continua o en momentos intermedios. Únicamente evaluación al principio y al final del bloque formativo o evaluación sumativa
- Momentos o actividades llave, que permitan la opción de varias vías de aprendizaje en paralelo, tales como itinerarios alternativos individuales o grupales
- Itinerarios adaptados a distintos perfiles. La clase se entiende como un todo, no existiendo adaptación ni individual ni de distintos subgrupos dentro de la clase

De todo ello se deduce que el modelo mixto o blended learning se incorpora en un nivel bajo de integración, utilizando los recursos online de manera escasa y básica y como fuente de información o comunicación, pero nunca como un elemento clave dentro del itinerario formativo. Del mismo modo, no se contempla el espacio online como fuente de trabajo y de ejecución de actividades complementario al presencial. Adicionalmente, destacan la ausencia de adaptación individual o grupal del método o del escenario y, por último, la unidimensionalidad del itinerario, no existiendo actividades en paralelo.

9.2.3. **Integración de recursos para ejecución de escenarios b-learning**

9.2.3.a. ***IMS Learning Design como método de modelado pedagógico***

Existen diversas especificaciones centradas en eLearning, como Scorm [300], Content Packaging [301] o Simple Sequencing [12], por citar algunas. De todas ellas, IMS Learning Design (o IMS-LD) [1] permite un modelado más flexible dependiente de los intereses del profesor o pedagogo y no de la intención de la especificación en sí. IMS-LD es una notación para representar cursos, lecciones o Unidades de Aprendizaje. Permite definir protagonistas, actividades, estructuras, momentos y un largo etcétera de elementos creados para potenciar el flujo de aprendizaje, como condiciones, propiedades o elementos de comunicación con ficheros XML externos [52; 53]. Pero, sobre todo, lo que le hace más útil es la posibilidad de definir el itinerario pedagógico según el modelo deseado por el diseñador/profesor y de adaptarlo a cada particularidad [302; 303].

Sin embargo, aunque IMS-LD es prometedor, queda lejos de ser una realidad en la práctica real docente, por diversos motivos [120]. Principalmente, la especificación nació en 2003 y no existe ninguna herramienta que permita la edición de alto nivel necesaria para creadores multidisciplinares. Es decir, que un profesor con base no técnica pueda modelar sus Unidades de Aprendizaje sin mayor conocimiento que la utilización habitual aunque específica de un editor. Además, aunque hay algún intento de visualizador como CopperCore Player [3], Reload Player [229] o Sled [42] no existe un sistema eficaz y con una instalación y una personalización sencillas para ejecutar las Unidades de Aprendizaje más complejas. Por último, no hay aún un sistema gestor de cursos o Course Management System – CMS que permita la integración natural de las unidades generadas. Con todo ello, los avances son muchos y, en menos de tres años, se ha pasado de ser un boceto de papel a contar con un grupo generoso de herramientas, grupos de investigación y proyectos [47].

Añadir que IMS-LD es también importante, como cualquier otra iniciativa open source o de código abierto, porque permite la reutilización del material generado para reuso particular o ajeno y porque facilita la edición y ejecución de una unidad de aprendizaje en cualquier sistema compatible con la especificación, sin necesidad de ceñirse a los habituales sistemas propietarios. En cierto modo, y para reforzar la argumentación, la evolución tiene cierta similitud con la desarrollada hace una década con las herramientas de edición multimedia y de diseño gráfico, pasando de formatos propietarios de cada casa (Macromedia, Adobe, Asymetrix, Kinetix…) a formatos inter-herramientas (PSD, FLA, DIR, 3DS…); también existe cierta cercanía con la evolución de las aplicaciones ofimáticas de hace dos décadas, pasando de modelos propietarios (WordStar, WordPerfect, Lotus 123…) a formatos inter-compatibles (RTF, DOC, XLS, Open Office…).

9.2.3.b. ***Moodle como entorno de creación y ejecución de cursos***

Por su parte, el sistema gestor de cursos o Course Management System – CMS Moodle [10] plantea desde una perspectiva de código abierto un modelo constructivista social, es decir, aprendizaje desde el conocimiento particular compartido con el grupo de usuarios. Mantiene, por tanto, un único enfoque de aprendizaje si bien, a través de los distintos recursos del sistema, se puede adaptar ligeramente a los deseos del profesor. Un curso se modela añadiendo temas y diversos elementos, como ficheros, imágenes, glosarios, wikis, foros y un largo etcétera. Al ser un sistema de código abierto con una acogida espectacular en el sector educativo (más de 100.000 usuarios en 150 países, 70 idiomas y 12.000 sitios registrados en el momento de escribir estas líneas) las ampliaciones y módulos de extensión se suceden con gran rapidez, pudiendo personalizarse ampliamente.

Moodle y IMS-LD tantean desde el año 2005 la posibilidad de integración [100]. En este sentido, Moodle y la Open University de Holanda mantienen un grupo de trabajo con el propósito de: a) Que un curso generado por Moodle pueda ser exportado a la notación de IMS-LD; b) que una unidad de aprendizaje generada bajo parámetros de IMS-LD pueda ser importada como curso dentro de un sistema Moodle; c) que una unidad IMS-LD pueda ser ejecutada dentro de Moodle. La implementación en tres pasos se hace necesaria para congeniar la necesidad que se percibe en el sector educativo con las restricciones en plazos y presupuestos propias del desarrollo.

9.2.3.c. ***Sled como visualizador de Unidades de Aprendizaje***

De los tres visualizadores mencionados (CopperCore Player, Reload IMS-LD Player y Sled) este último permite la ejecución remota de Unidades de Aprendizaje ejecutadas bajo un motor CopperCore. Es decir, un cliente a través de un visualizador web habitual (por ejemplo, Microsoft Internet Explorer) visualiza unidades almacenadas en un servidor que ejecuta el motor. Sled permite también una fácil personalización y extensión, al estar diseñado en código abierto. Por el contrario el grado de personalización de los otros dos visualizadores existentes es más complicado y también son players dedicados, debiendo instalarse en el cliente como utilidad independiente. Sled es, por tanto, más versátil y fácilmente ejecutable.

9.2.4. **Propuesta de integración técnica**

Teniendo en cuenta modelos anteriores de integración de paquetes de información en aplicaciones de ejecución o visualización, como SCORM dentro de un visualizador Reload [72]. El modelo que proponemos a continuación trata de subsanar el hueco existente entre unidad de aprendizaje, sistema gestor de cursos y visualizador.

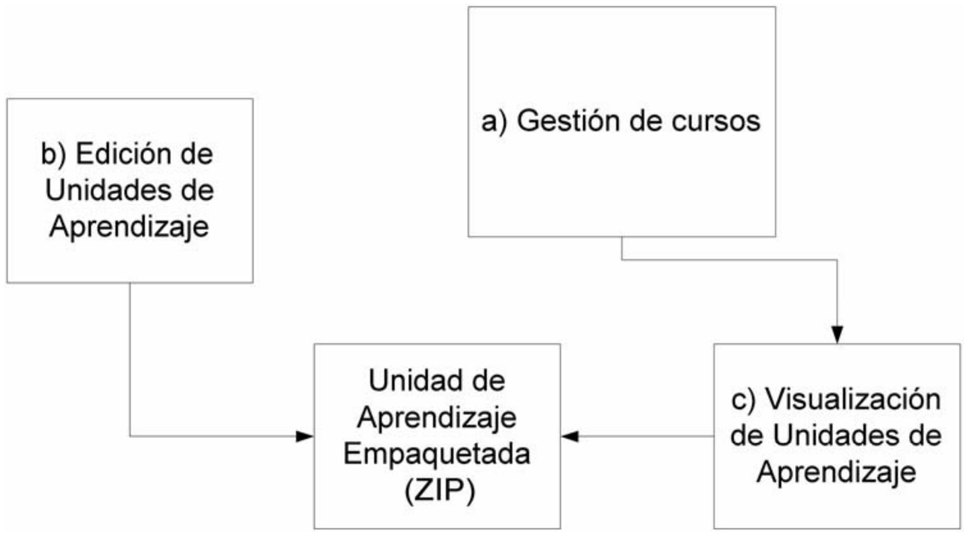


Figure 70. Integración con llamada externa a visualizador

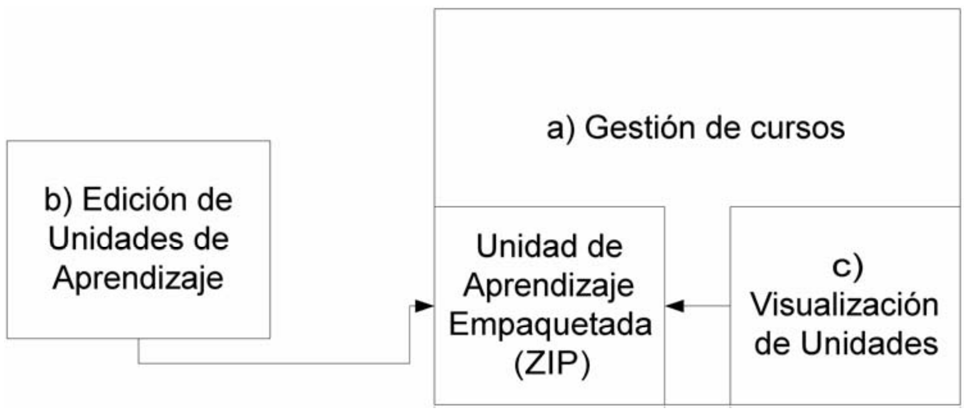


Figure 71. Integración con llamada interna a visualizador

Las ilustraciones Figure 70 y Figure 71 muestran las distintas capas del sistema integrado: a) Por un lado, el sistema gestor de cursos realiza toda la administración de usuarios y de cursos; b) por otro lado, un editor de IMS-LD permite la generación de Unidades de Aprendizaje siguiendo la especificación; c) por último, un enlace de Moodle llama al visualizador Sled pudiendo ejecutar la unidad generada en IMS-LD. Una modificación de este apartado c) sería que la unidad fuera ejecutada dentro del sistema gestor de cursos, es decir que existiera un visualizador integrado plenamente en Moodle, en lugar de llamarlo mediante un enlace externo.

De esta manera, los cursos podrían generarse siguiendo la especificación, lo que garantizaría la reutilización y la interoperabilidad de los contenidos y, sobre todo, de la metodología de aprendizaje, y asimismo podrían ejecutarse de manera remota mediante un visualizador web (Internet Explorer) que se llamaría desde el gestor de cursos (Moodle), completando el círculo. En esta propuesta de sistema integrado se daría pie a cubrir los puntos críticos detectados en el apartado práctico de este artículo, permitiendo la incorporación del sistema online como un todo constituyente y clave de la unidad de aprendizaje mixta y no únicamente como un fondo de recursos en Internet, paralelo y complementario al discurso casi único presencial. La OUNL ha realizado diversas pruebas sobre esta configuración que demuestran su viabilidad.

9.2.5. **Conclusiones**

Existe una diferencia entre la fundamentación teórica y la práctica docente relativa a escenarios de aprendizaje en entornos mixtos o blended learning (presencia-online). Diversos escenarios que se utilizan en el día a día no encuentran una referencia en la literatura, lo que origina que se consideren de manera aislada en vez de cómo parte de una tendencia o incluso de un patrón de enseñanza/aprendizaje. La recopilación y categorización de estos patrones conlleva la necesaria reutilización e interoperabilidad del material educativo, lo que rentabiliza el tiempo y el esfuerzo del profesor y permite la actualización y adaptación del contenido pedagógico y de la metodología empleada a los estudiantes o grupos de estudiantes. En este sentido, fruto de dos casos prácticos realizados con setenta profesores universitarios en activo, se han obtenido una serie de conclusiones que invitan a pensar que la concepción y ejecución de escenarios blended está por debajo de las posibilidades reales actuales y que no explota algunos conceptos innovadores, como la ejecución de itinerarios paralelos o la adaptación individual o grupal de los mismos.

A través de las herramientas y mecanismos actuales existentes, se propone la integración de un sistema gestor de cursos (Moodle), un visualizador de Unidades de Aprendizaje (Sled) y una especificación eLearning (IMS Learning Design) para agilizar y materializar un entorno que permite la edición, gestión y ejecución de Unidades de Aprendizaje y de grupos de aprendizaje de manera sencilla y útil. Bien es cierto que la existencia de un sistema único con todo integrado bajo una misma plataforma open source facilitaría la incorporación de este tipo de enfoque a la práctica real de enseñanza mixta, y que aumentaría la extensión del software mediante módulos adicionales y personalizaciones (como ocurre actualmente con Moodle y otros sistemas abiertos de gran calado) y la divulgación de la metodología de generación de material docente. Aun así, la demanda imperiosa de la comunidad educativa por encontrar al menos una nueva vía que tienda a congeniar los esfuerzos productivos con las restricciones de creación y ejecución, hace que la propuesta integradora presentada sea un primer paso real a tener en cuenta.

9.3 . Representation of coordination mechanisms in IMS-LD

This section is focused on the Representation of Coordination Mechanisms in IMS Learning Design to Support Group-based Learning and it is co-authored by Yongwu Miao (Open University of The Netherlands) and Dai Griffiths (University of Bolton) [304].

Group interaction has to be meticulously designed to foster effective and efficient collaborative learning. The IMS Learning Design specification (IMS-LD) can be used to create a formal representation of group interaction and the model can then be used to scaffold group interaction by means of coordination support at run-time. In this chapter, we investigate the expressiveness of IMS-LD in representing coordination mechanisms by using coordination theory as an analytical framework. We have found that IMS-LD can represent almost all the basic coordination mechanisms. We have also identified some hurdles to be overcome in representing certain coordination mechanisms. According to coordination theory, common coordination mechanisms can be reused in different settings. We briefly explore the feasibility of representing coordination mechanisms at a high-level of abstraction, which will be easier for instruction designers and teachers to understand and use.

9.3.1. Introduction

Group-based learning is an instructional strategy that provides a group of learners with intensive group interaction that can deepen individual learners' understanding. Well-organized group-based learning may result in collaboratively produced knowledge objects or conceptual artifacts which could not be created by any individual learner in the group acting alone. However, the benefits of this instructional strategy have a cost, because additional coordination activities have to be carried out while learners perform learning activities. Examples of such coordination activities are allocating tasks, distributing and exchanging information, and managing work sequences. Although coordination activities do not directly contribute to the production of knowledge objects or conceptual artifacts, they have an influence on the effectiveness and efficiency of group-based learning, and sometimes on its success or failure.

In face-to-face learning rich communication channels are available to support group interaction. These are lost in computer-based learning, and so in this environment there is a need to provide computational coordination mechanisms. One promising technical solution is to provide a formal model of a well-designed group interaction by using a process modeling language, and then to coordinate learners' interactions according to this model in a language-compatible execution environment. This enables learners to focus on learning activities without having to pay too much attention to coordination problems, and so supports enhanced effectiveness and efficiency of group-based learning in computer-based environments.

IMS Learning Design [1] is an educational process modeling language which can be used to model a wider range of pedagogical strategies, including collaborative learning [85]. A basic introduction to IMS-LD is available in the chapter (Using the IMS-LD Standard to Describe Learning Designs, Koper and Miao, this book). The purpose of this present chapter is to systematically investigate the expressiveness of IMS-LD in representing coordination mechanisms which support group interaction, and the approach taken is to use coordination theory as an analytical framework. We also provide XML (Extensible Markup Language) code, to illustrate how group interaction can be represented in IMS-LD.

It is important to note that characteristics of group-based learning processes vary from well-structured to highly fluid. Highly fluid collaborative processes, in which it is unpredictable who will take which action when, and how other group members will respond, are not well suited to coordination using computational mechanisms. The attempt to specify a fluid collaborative process in detail often raises the so-called "over-scripting" problem [305], which may restrict group interaction to some extent. Some fluid collaborations are suited to coordination by human users. These may be defined in IMS-LD, for example, as a collaborative activity with a conference service (e.g., an audio/video conferencing, text-based chat tool, or a discussion forum). The users (e.g., tutors and students) are expected to solve their coordination problems by using functions offered by the service. It may be seen that using this approach the coordination within an activity is not specified at the process level in the learning design, and that responsibility for process control is shifted to the user at execution time. This is, therefore, outside the scope of this chapter, which focuses on how computational mechanisms can be represented in IMS-LD.

9.3.2. **Background**

This section briefly introduces group-based learning and *coordination theory*.

9.3.2.a. **Group-based Learning and Collaboration Scripts**

Learning in small groups has been intensively researched since the 1970s. According to Tribe [306] there are two main types of purpose for group-based learning in higher education: those related to skills acquisition and those related to academic aims. As Tribe [306] summarized, the skills acquired in group-based learning cover such interpersonal competences as oral communication, active listening, group leadership, group membership, the ability to examine assumptions, and the ability to tolerate ambiguities. All of these skills are highly valued in employment. The academic objectives which build on these employment skills include the ability to understand a text, question a line of argument, follow up a lecture, and gauge an individual's progress on a particular course or evaluate a course.

According to [307; 308] there is agreement on five components of 'group-based learning'. As Strijbos el. al. summarize, firstly, groups are composed of either a minimum of two up to six participants. Secondly, group-based learning is characterized by 'positive interdependence', which refers to the degree to which the performance of a single member is dependent on the performance of all others [309]. A third component is the task, which must be a genuine group task, in which the effort of all group members is needed. A fourth component is 'individual accountability'. This refers to each student's individual responsibility for a specific aspect of the group process or group performance (or both). Individual accountability is enhanced through grading students for their individual effort or performance, as well as the group's performance. The fifth and final component is a shift from 'teacher centered' to 'student centered'.

Early studies on group-based learning focused on the role of independent variables that might influence the learning outcome, e.g. group size and group dynamics. Recent studies, however, analyse group interactions in order to ground the design of the support to be provided. According to Dillenbourg [305], the key to understanding collaborative learning is to gain an understanding of the interactions among individuals. Recently in the Computer-supported collaborative learning (CSCL) community, the design of collaboration scripts has been a new focus area. The basic idea is to formally describe group interaction by using a scripting language and then to coordinate group members and their actions by executing collaboration scripts [310-315]. Some efforts [255; 313; 316]) have been made to investigate whether IMS-LD is sufficiently expressive to represent collaborative learning processes effectively, usually by analyzing special cases. The most serious research in this direction was done by Van Es and Koper [255], which investigated many examples, randomly selected from 6034 lesson plans. In the research described in this chapter, not only a case study method (the case used here is mainly for the purpose of explanation), but also a theory-based analysis method is adopted to systematically test the capacity of IMS-LD in representing coordination mechanisms.

9.3.2.b. **Coordination Theory**

Coordination theory concerns the interdisciplinary study of coordination, which is defined as the process of managing dependencies between activities. Malone and Crowston [317] analyzed processes in terms of actors performing interdependent tasks. These tasks might require or create resources of various types. Coordination theory provides a theoretical framework for analyzing coordination in complex processes, thus contributing to user task analysis and modeling. It has been applied in many fields, including computer science, organization theory, economics, management science, sociology, social psychology, anthropology, linguistics, law, political science, and so on. The research reported here is the first time that coordination theory has been applied to education.

One of the most powerful contributions of coordination theory is to systematically identify and analyze a wide variety of *dependencies*. Three elementary dependency types are identified in coordination theory: 1. *Sharing*, 2. *Flow*, and 3. *Fit*. In sharing dependencies two or more activities share the same resource(s). *Sharing dependency* frequently occurs when one resource is used by a number of people or activities, whether that resource is a machine on a factory floor, a budget, or a room, or anything else which is used in multiple activities. In *flow dependencies* resources produced by one activity are consumed by one or more subsequent activities. The concept of flow is intuitive and ubiquitous, emerging from the succession of events in human activity. In *fit dependencies* two activities concurrently produce the components of the same resource, and these have to fit together. A good example of *fit* is the design of a car, where one engineer designs the engine, another designs the body, and so forth. Dependencies arise between the activities because all the parts have to fit together in the same car.

It is important to note that these three dependency types can be further specialized. For example, the flow dependency can be divided into three sub-dependencies: *precedence*, *transfer* and *usability*. *Precedence dependency* indicates that the actor performing the second task has to know when the resource is available and the task can be started. *Transfer dependency* indicates that the resource must be moved from the activity in which it was created to the activity in which it is consumed. Finally, *usability dependency* indicates that the resource created by the first task must be appropriate for the needs of the second task. The fit dependency can be further specified as a *decomposition dependency* between task and sub-task.

According to coordination theory, all dependencies in any relationship can be analyzed as either combinations of, or more specialized types of, these three elementary types or their sub-types. The theory describes how these dependencies can present actors in organizations with *coordination problems* which constrain the efficiency of task performance. To overcome coordination problems, actors must perform additional activities such as allocating tasks and control workflow and information-flow, which Malone and Crowston called *coordination mechanisms* or *coordination activities* [317]. Many such mechanisms to manage dependencies have been identified in organizations.

Different organizations which have similar goals and achieve them using more or less the same set of coordination activities will have to manage the same dependencies. Nevertheless, they may choose to use different coordination mechanisms, thus resulting in different processes [318]. The best process to use depends on situational factors and often involves trade offs.

9.3.3. Representation of coordination mechanisms in IMS-LD: a case study of group_based learning

In this section, based on coordination theory, we analyze the coordination problems which arise in group-based learning processes, and also systematically explore the degree to which IMS-LD can represent possible coordination mechanisms for supporting group interaction, either directly or indirectly. The investigation is conducted and explained using the "Knowledge Convergence Script" use case, which is briefly introduced at the beginning of this section.

9.3.3.a. Knowledge Convergence Script

We have chosen to model an example of group-based learning which is well documented in the literature [319]. This was conducted in a web-based environment, with a small group of three learners who were required to write three reports about three cases. Following the original design the whole process is carried out in four stages.

1. *Case reporting*: each learner reads a different case and writes a report about the case which they have read. When all three learners have finished their reports, they pass them on to designated co-learners the first round of a pre-defined pattern of rotation.
2. *Criticizing 1*: each learner comments on the report which they have received. When all three have finished the first round of comments, they rotate the reports again, together with the first round comments.
3. *Criticizing 2*: each learner comments on the newly transferred report and the associated comment. When all three have finished the second round of comments, they rotate the reports again, together with the first and second round comments.
4. *Finalizing the report*: each report returns to the original author together with two comments. Each learner revises their own report (writes a synthesis to merge the ideas of other learners) in light of their comments.

The "Knowledge Convergence Script" has been implemented in a web-based collaborative learning environment, and it is reported that this group-based learning strategy is effective and efficient [319]. In supporting this group-based learning strategy we use process modeling and execution approach, rather than a software development approach. Figure 72 illustrates the process model, using the following conventions:

- Light-green rectangles represent stages
- Orange rectangles represent activities
- White rectangles represent artifacts
- Solid arrows indicates workflows
- Dashed arrows indicate information-flows.

Three learners are shown: *learner1*, *learner2*, and *learner3*, who work through a four-stage work procedure including *Case reporting*, *Criticizing 1*, *Criticizing 2*, and *Finalizing the report*. At each stage, three learners perform activities in parallel to produce artifacts which will be used as input of succeeding activities carried out by their peers. For example, at the first stage *learner1* performs activity *reporting1*. He/she reads *Case1* and produces artifact *InitialReport1*, which is then transferred to the activity *criticizing2-1* at the second stage. *Learner2* produces artifact *comment2-1*, which is then transferred together with *Case1* and *InitialReport1* to *learner3*. At the third stage *learner3* reads *Case1*, *InitialReport1*, and *Comment2-1* and writes *Comment3-1*. Finally, all documents associated with *Report1* are transferred to *learner1*. He/she improves *Report1* based on the received comments, and then produces a final version of the case report *FinalReport1*.

We use IMS-LD to specify this strategy in the form of XML. The resulting model [320] can be executed in any IMS-LD compliant run-time environment, such as CopperCore [3]. Figure 73 shows a screenshot of CopperCore used to run this script when learner1 is writing the final report. The top-left pane shows the work procedure of the user. The bottom-left part shows all environments associated with the activity currently being performed, which include the documents to be accessed by the user. When the user clicks a learning object (such as a case and a comment made available in the environment), the content of the learning object is presented in the right part of the window. In the screen shown below the user has selected the final activity write final report. The main area of the window presents the activity-description of write final report activity, in which the user writes the final version of his/her case report as shown in Figure 73. It is important to note that the main goal of this research is not to study whether this group-based learning strategy is effective or efficient, but to investigate and demonstrate the expressiveness of IMS-LD in modeling group-based learning strategy. Moreover, we observe that various group-based learning strategies can be adopted to achieve the same learning goal, and that no single strategy is ideal for all situations. Accordingly we designed some alternatives, which are not intended to improve this group-based learning process, but rather to provide the basis for a discussion of possible coordination mechanisms.

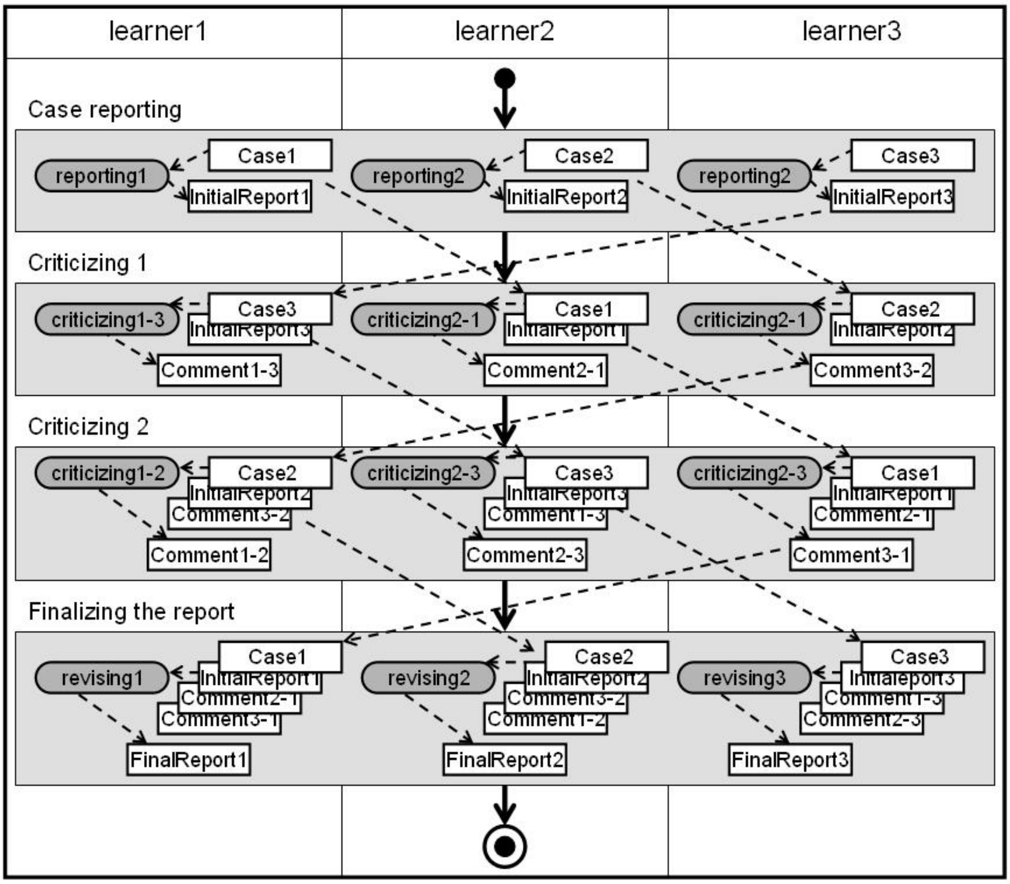


Figure 72. An activity diagram of "Knowledge Convergence Script"

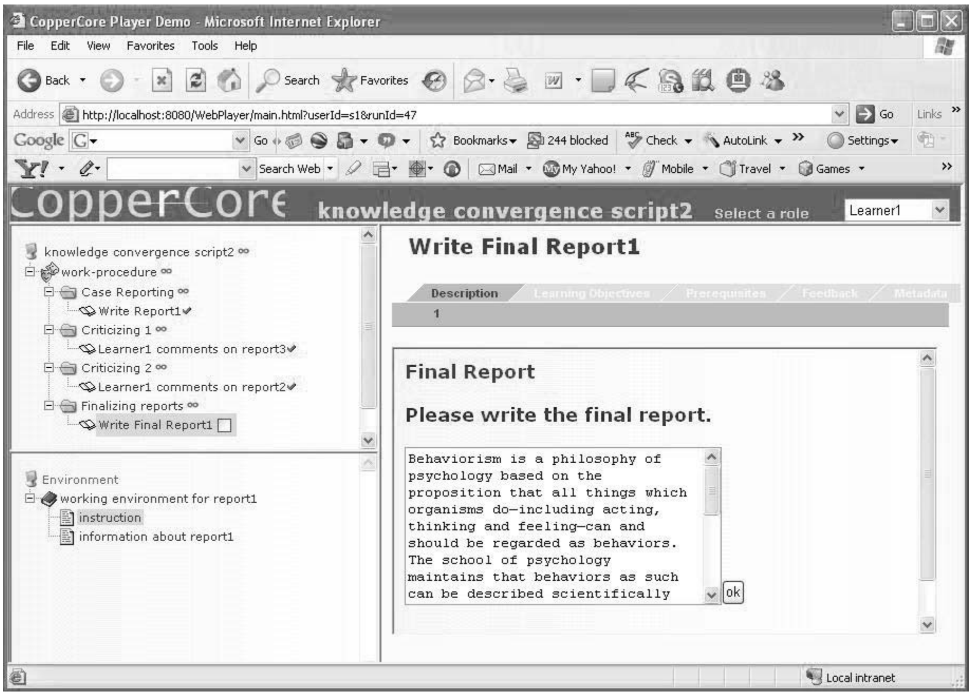


Figure 73. A screenshot of CopperCore running the "Knowledge Convergence Script"

9.3.3.b. **Analyses of dependencies and possible coordination mechanisms in group-based learning processes**

In this section, we investigate various forms of dependences in group-based learning processes from the perspective of coordination theory. The "Knowledge Convergence Script" and its alternatives are used as examples to analyze and explain the coordination mechanisms to manage various dependencies in group-based learning processes.

Sharing dependencies

In the activities carried out in this use case each learner has to read three cases and make contributions to each report. Thus the resources shared by the activities are three learners and three cases. If learners work without any coordination mechanism for managing the sharing dependencies described above, the result will be disorder, with each learner performing any task at any time.

The coordination mechanism used in the original design is to pre-define the allocation of learners and cases to activities, so that some are carried out concurrently, and others at different times. In order to support this coordination mechanism it is necessary to represent the bindings between the actors and the activities which they will carry out, either concurrently or at different times. It is also necessary to ensure that actors have access to the appropriate cases when they have to carry out a particular activity. Another possible coordination mechanism for managing sharing dependencies is that three learners and one case will be allocated to individual activities in turn. Each activity is itself a collaborative task. In order to support this coordination mechanism, it is necessary to represent the binding between the multiple actors and the same activity at the same time, and to represent the use of the communication tools used to exchange their ideas and create the report.

These two strategies are static coordination mechanisms, which manage the sharing dependencies in a pre-defined manner. If it is not decided in advance which learner will be responsible for reporting on which case then a dynamic coordination process will be required which responds to the dynamics of the learning process. An example of a dynamic coordination mechanism is "first come, first served", and this mechanism can be applied to determining the pattern of rotation. For example, we could add a register activity for each role, but not allocate any activity to any role in design time. At run-time, three users will register to carry out the process, and according to the sequence of their registrations, the activities will be allocated and the artifacts rotated.

Flow dependencies

As mentioned before, the flow dependency has three sub-dependencies: 1) *precedence*, 2) *transfer* and 3) *usability*. We now analyze these types of dependencies in the group-based learning.

1) *Precedence*: in the use case, there are precedence dependencies between some activities. For example, when one learner has finished the activity of creating an initial report, the other two can comment on it in turn. Only after other two learners provide their comments can the first learner write a synthesis.

Normally, the coordination mechanism used to manage precedence dependencies is event-driven. This means that an event (e.g., the termination of an activity and the available of a resource) triggers the start the succeeding activity. In complex learning process, branching, forking, and joining are possible coordination mechanisms. Branching means a control that only one succeeding activity will be triggered among several candidates according to a condition. Forking refers to the control that two or more succeeding activities will start in parallel after the termination of an activity. Joining is a control that the termination of all preceding activities triggers the start of a succeeding activity.

In the original “Knowledge Convergence Script” design, a four-step process is used. In each phase, three activities are performed in parallel. Only when all activities in one step are finished will all the activities in the next step be triggered. This is a synchronization coordination mechanism. However, if the concurrent tasks performed within a step are not balanced, the efficiency of this coordination mechanism is not high. For example, if one of the three cases is more difficult and takes longer to understand and to develop ideas, then at each step the activity handling this case will take longer. Using synchronization each step takes as long as the most difficult case takes to resolve.

In order to enhance the efficiency a task-driven approach can be used, so that when a learner finishes the current task s/he can perform her/his succeeding activity without having to wait. When there are unbalanced tasks this coordination mechanism can reduce the total learning time. Another possible coordination mechanism is to trigger an activity by an event indicating that all necessary resources are available (data-driven). For example, each learner is responsible for performing four activities: creating an initial report, commenting on two other reports, and writing a final report. Using this approach whenever an initial report written becomes available, the corresponding activity for commenting on it is triggered, even if the learner who will carry it out is still working on her/his initial report.

2) *Transfer*: in group-based learning, an artifact is usually employed as a means of coordinating group interaction and constitutes a collaboratively produced knowledge object. In the use case, there are transfer dependencies between some activities. For example, artifacts such as initial reports and comments produced in an activity are transferred to other activities.

The basic coordination mechanism for managing transfer dependencies is to capture the artifact produced in the activity and to present the captured artifact in other activities.

3) *Usability*: in the use case, there are usability dependencies between activities. For example, an initial report of a case should transfer to an activity which has the aim of commenting on this report.

As mentioned above, in e-learning processes, the objects to be transferred are information objects. The coordination mechanisms for managing usability dependencies should check whether the class of the artifacts, data type, size, and other constraints meet the requirements.

Fit dependencies

In the use case each final report is a synthesis of ideas from all the learners, while the production of each report is split into four activities. The use case could be extended so that the three cases are specified as behaviorism, cognitivism, and constructivism, with the three reports being assembled into a general report about learning theories. In this extended case the activities of writing the three reports would have fit dependencies.

A basic coordination mechanism for managing fit dependencies is to check whether the classes of the artifacts, data types, sizes, and other constraints are compatible. A basic coordination mechanism for managing decomposition dependencies between task and sub-task is also needed.

9.3.3.c. Representation of coordination mechanisms in IMS-LD

In this section, we analyze whether IMS-LD can represent the coordination mechanisms for managing various dependencies within group-based learning processes which we have identified above.

There are two kinds of activities defined in IMS-LD: learning activity and support activity. It is not necessary to distinguish them for our present purpose, and so we simply use the term activity. The notations representing resources in IMS-LD are role, environment, learning object, and learning service. For the sake of clarity, we discuss these in turn for fit, flow, and sharing.

Representation of Coordination Mechanisms to Manage Fit Dependencies

IMS-LD has no notation which explicitly represents artifacts, and so no computational coordination mechanism is available to check whether the components of an artifact fit together. In IMS-LD, a general notation property can be used to represent a variety of concepts including artifacts created in the learning processes. Depending on its scope, an artifact can be defined as a global property or a local property (run property). Similarly, an artifact can be defined as a personal property, a role property, or a general property, depending on its owners. A property cannot represent complex, structured information objects because it can only have a primitive data type such as integer, real, string text, URL, file, time, and so on. Consequently IMS-LD provides no computational mechanism for coordinating the assembly of components produced simultaneously in different activities. As shown in the use case, the merging work is performed without computational support. Of course, as a general process modeling language, IMS-LD should not and cannot directly support any specific artifact. One possible solution is to use a file type suitable for the representation of structured information (e.g., XML files). If external learning services were integrated which checked and assembled components and handled specific artifacts, then the IMS-LD engine could communicate with these mechanisms in order to manage the specific fit dependencies. This is a complicated technical issue, however, and so we do not discuss it in detail in this chapter.

Although IMS-LD can only manage artifact decomposition dependencies indirectly, it provides several coordination mechanisms which can be used to directly manage task decomposition dependency. In IMS-LD a learning process can be decomposed into plays, acts, and role-parts. Each role-part consists of a role and an activity or an activity-structure that is recursively decomposable. All these notations can be used to represent a set of tasks with a variety of granularities as a hierarchical structure. However, the restriction to activity-structure in which all activities have to be performed by the same role makes it inconvenient to represent a sequence of activities performed, for example, by different roles in turn. If IMS-LD had a construct corresponding to a role-part-sequence, then it would be easier to represent a group-interaction sequence involving various roles.

In IMS-LD, a role can be decomposed into sub-roles at arbitrary levels. For each role, some attributes can be used to restrict the role, such as max-members, min-members, inclusive/exclusive, and so on. However, no constraint specifies how a role should be composed of sub-roles. As a result it is sometimes difficult to define the formation of a group when it is modeled using role notation. For example, if a group must be formed by three (two female and one male) learners with backgrounds in pedagogy, psychology and computer science respectively, it is difficult to represent such a constraint in IMS-LD. As a consequence, no computational mechanism can be used to check whether the group has been correctly formed. We do not go into this in greater detail because there is no a simple method to resolve the issue, and in any event the case under discussion does not raise this particular problem.

In order to provide clarify how to model group interaction in IMS-LD without going into too much technical detail, we now introduce a restricted pseudo-code, based on IMS-LD. Figure 74 illustrates some definitions of the structure of roles, properties representing artifacts, and activity decompositions. Figure 74a defines three learners: learner1, learner2, and learner3. The constraints for each role are that one and only one user can play a role, and a user cannot have more than one role in this process. The code shown in Figure 74b specifies several properties InitialReport1, Comment1-2, Comment1-3, and FinalReport1, which represent artifacts produced by learner1. Figure 74c defines four activities performed by learner1. Each activity will be carried out in an associated environment. Note that the corresponding set of properties and activities relevant to learner2 and learner3 are omitted.

```
    <learner create-new="not-allowed" identifier="learner1" match-
persons="exclusively-in-roles" max-persons="1" min-persons="1">
        <title>Learner1</title>
    </learner>
    <learner create-new="not-allowed" identifier="learner2" match-
persons="exclusively-in-roles" max-persons="1" min-persons="1">
        <title>Learner2</title>
    </learner>
    <learner create-new="not-allowed" identifier="learner3" match-
persons="exclusively-in-roles" max-persons="1" min-persons="1">
        <title>Learner3</title>
    </learner>
```

Figure 74a: the definitions of three roles

```
<!– the definition of the property representing the initial report written
by learner1 -->
    <loc-property identifier="InitialReport1">
        <title>Initial Report1</title>
    </loc-property>
<!– the definition of the property representing the comment written by
learner1 on the initial report2 written by learner2 -->
    <loc-property identifier="Comment1-2">
        <title>Comment1-2</title>
    </loc-property>
```

```
<!— the definition of the property representing the comment written by
learner1 on the initial report3 written by learner3 -->
    <loc-property identifier="Comment1-3">
        <title>Comment1-3</title>
    </loc-property>
<!— the definition of the property representing the final report written by
learner1 -->
    <loc-property identifier="FinalReport1">
        <title>Final Report1</title>
    </loc-property>
```

Figure 74b: the definitions of properties relevant to learner1

```
<!—the definitions of an activity arranged for learner1 to write a case
report -->
<learning-activity identifier="LA-write-initial-report1">
    <title>Write Report1</title>
    <environment-ref ref="ENV-for-report1"/>
    <activity-description>
        <title>Write Report</title>
        <item identifier="ITEM-write-report1" identifierref="RESO-write-
report1" />
    </activity-description>
</learning-activity>
<!—the definitions of an activity arranged for learner1 to comment on the
InitialReport2 written by learner2 -->
<learning-activity identifier="LA-comment-1-2">
    <title>Learner1 comments on report2</title>
    <environment-ref ref="ENV-for-report2"/>
    <activity-description>
        <title>Commenting</title>
        <item identifier="ITEM-write-comment-1-2" identifierref="RESO-
comment-1-2" />
    </activity-description>
</learning-activity>
<!—the definitions of an activity arranged for learner1 to comment on the
InitialReport3 written by learner3 -->
<learning-activity identifier="LA-comment-1-3">
    <title>Learner1 comments on report3</title>
    <environment-ref ref="ENV-for-report3"/>
    <activity-description>
        <title>Commenting</title>
```

```
        <item identifier="ITEM-write-comment-1-3" identifierref="RESO-
comment-1-3" />
    </activity-description>
</learning-activity>
<!—the definitions of an activity arranged for learner1 to write the
FinalReport1 -->
<learning-activity identifier="LA-write-final-report1">
    <title>Write Final Report1</title>
    <environment-ref ref="ENV-for-report1"/>
    <activity-description>
        <title>Write Final Report</title>
        <item identifier="ITEM-AD-write-final-report1" identifierref="RESO-
write-final-report1" />
    </activity-description>
</learning-activity>
```

Figure 74c: the definitions of activities relevant to learner1

Figure 74. The definitions of roles, properties, activities

Representation of Coordination Mechanisms to Manage Flow Dependencies

Precedence: IMS-LD provides several built-in mechanisms to manage the precedence dependencies, such as acts in a play and activity-structure with a sequence type. Note that such sequences are weak coordination mechanisms, because the sequences are no more than suggestions. The users can work following the sequences or vary them, because all acts and activities are accessible at any time. They can even access completed activities. This has advantages, because it provides flexibility for the users to carry out tasks as they wish. It is sometimes difficult to judge if an activity has really terminated, especially in learning processes. For example, when learners work on reading and understanding an article and after a period of time they think the task has been finished, they terminate the activity and move on to the next one. However, they may recognize that they did not fully understand the article and go back to read it again. Weak sequence control mechanisms make it possible for users to carry out such tasks flexibly and handle exceptions manually. On the other hand, users have to pay attention to coordination problems, to a greater or lesser extent. Moreover, such freedom for users to decide the actual work sequence may create problems, especially in situations where a strictly defined route is required. Fortunately, IMS-LD provides additional mechanisms to support strong controls for sequence of acts and sequence of activities. The following paragraphs will present how weak and strong sequencing mechanisms can be represented in IMS-LD.

As shown in Figure 75, the work procedure of this group-based learning is modeled as four acts titled *Case Reporting*, *Criticizing 1*, *Criticizing 1*, and *Finalizing reports*. Each act represents a stage, in which who is responsible for doing which activity is specified as a role-part. In the first act titled *Case Reporting*, for example, *learner1* is assigned to perform the activity titled *Write report1*, which is defined in Figure 74 using identifier *LA-write-initial-report1*. Using a weak sequencing mechanism, we can represent four acts in sequence without control as shown in Figure 74. However, it is possible to represent a strong sequencing mechanism in IMS-LD in a way to specify the completion condition for an act. One such a condition is that an act will be terminated automatically by the system when all role-parts in the act are completed. For example, the first act completes when all learners finish the activities to create their initial reports, and then the activities in the succeeding act titled *Criticizing 1* become accessible.

```
<!—the definitions of four acts in a play -->
<play identifier="PL-work-procedure">
    <title>work-procedure</title>
    <!—the definitions of the first act-->
    <act identifier="ACT-case-reporting">
        <title>Case Reporting</title>
        <role-part identifier="RP-write-report1">
            <title>learner1 writes report1</title>
            <role-ref ref="learner1"/>
            <learning-activity-ref ref="LA-write-initial-report1"/>
        </role-part>
        <role-part identifier="RP-write-report2">
            <title>learner2 writes report2</title>
            <role-ref ref="learner2"/>
            <learning-activity-ref ref="LA-write-initial-report2"/>
        </role-part>
        <role-part identifier="RP-write-report3">
            <title>learner3 writes report3</title>
            <role-ref ref="learner3"/>
            <learning-activity-ref ref="LA-write-initial-report3"/>
        </role-part>
    </act>
    <!—the definitions of the second act -->
    <act identifier="ACT-criticizing1">
        <title>Criticizing 1</title>
        <role-part identifier="RP-comment-1-3">
            <title>learner1 comments on report3</title>
```

```
            <role-ref ref="learner1"/>
            <learning-activity-ref ref="LA-comment-1-3"/>
        </role-part>
        <role-part identifier="RP-comment-2-1">
            <title>learner2 comments on report1</title>
            <role-ref ref="learner2"/>
            <learning-activity-ref ref="LA-comment-2-1"/>
        </role-part>
        <role-part identifier="RP-comment-3-2">
            <title>learner3 comments on report2</title>
            <role-ref ref="learner3"/>
            <learning-activity-ref ref="LA-comment-3-2"/>
        </role-part>
    </act>
    <!—the definitions of the third act -->
    <act identifier="ACT-criticizing2">
        <title>Criticizing 2</title>
        <role-part identifier="RP-comment-1-2">
            <title>learner1 comments on report2</title>
            <role-ref ref="learner1"/>
            <learning-activity-ref ref="LA-comment-1-2"/>
        </role-part>
        <role-part identifier="RP-comment-2-3">
            <title>learner2 comments on report3</title>
            <role-ref ref="learner2"/>
            <learning-activity-ref ref="LA-comment-2-3"/>
        </role-part>
        <role-part identifier="RP-comment-3-1">
            <title>learner3 comments on report1</title>
            <role-ref ref="learner3"/>
            <learning-activity-ref ref="LA-comment-3-1"/>
        </role-part>
    </act>
    <!—the definitions of the final act -->
    <act identifier="ACT-finalizing-report">
        <title>Finalizing reports</title>
        <role-part identifier="RP-write-final-report1">
            <title>learner1 writes the final report</title>
            <role-ref ref="learner1"/>
            <learning-activity-ref ref="LA-write-final-report1"/>
        </role-part>
```

```
        <role-part identifier="RP-write-final-report2">
            <title>learner2 writes the final report</title>
            <role-ref ref="learner2"/>
            <learning-activity-ref ref="LA-write-final-report2"/>
        </role-part>
        <role-part identifier="RP-write-final-report3">
            <title>learner3 writes the final report</title>
            <role-ref ref="learner3"/>
            <learning-activity-ref ref="LA-write-final-report3"/>
        </role-part>
</act></play>
```

Figure 75. The definition of a sequence of acts in a play

In order to support strong precedence dependencies between activities, we can represent the sequence by using conditions to set the visibility of activities. Figure 76 shows an example which supports strong precedence dependency between two activities using a condition. As shown in Figure 76, if and only if the first activity, which identifier is *LA-write-initial-report1*, is completed, the second activity, which identifier is *LA-comment-1-3*, becomes accessible. Meanwhile, the first activity becomes inaccessible unless it is specifically set to be visible in other conditions.

```
<if>
    <complete>
        <learning-activity-ref ref="LA-write-initial-report1"/>
    </complete>
</if>
<then>
    <hide>
        <learning-activity-ref ref="LA-write-initial-report1"/>
    </hide>
    <show>
        <learning-activity-ref ref="LA-comment-1-3"/>
    </show>
</then>
<else>
    <hide>
        <learning-activity-ref ref="LA-comment-1-3"/>
    </hide>
</else>
```

Figure 76. The definition of a condition managing a strong precedence dependency between two activities

The coordination mechanisms discussed above for managing precedence dependencies are task-driven mechanisms. In IMS-LD conditions can also be used to represent data-driven mechanisms. For example, if *learner1* submits his/her initial case report, *learner2* can start to perform the activity (its identifier is *LA-comment-2-1*). Otherwise, this activity will be kept hidden from its actor. Figure 77 illustrates this example.

```
<if>
    <not>
        <no-value>
            <property-ref ref="InitialReport1"/>
        </no-value>
    </not>
</if>
<then>
    <show>
        <learning-activity-ref ref="LA-comment-2-1"/>
    </show>
</then>
<else>
    <hide>
        <learning-activity-ref ref="LA-comment-2-1"/>
    </hide>
</else>
```

Figure 77. The definition of a condition representing a data-driven coordination mechanism

Transfer: IMS-LD has no notation which explicitly represents the transference of an artifact produced in an activity and consumed by other activities. However, the transference of an artifact can be represented indirectly. Figure 78 shows an example which transfers an initial report created by *learner1* in the activity *Write Report1* to the activity *Learner2 comments on report1*. Figure 78a shows the definition of the first activity *Write Report1*, in which learner1 writes initial *report1* using the information item *ITEM-write-report1* that refers to a resource *RESO-write-report1*. Figure 78b shows the content of resource file *RESO-write-report1*, in which a global element *set-property* is used to input the initial *report1* captured by the property *InitialReport1*. Figure 78c defines the second activity titled *Learner2 comments on report1* which is associated with the environment *ENV-for-report1*, defined in Figure 78d. This environment contains a learning object *LO-information-about-report1*, which has an information item *ITEM-report1*. This item refers to the resource *RESO-presentation-of-report1* and it will become visible when the *InitialReport1* is made available. Figure 78e shows the content of resource file *RESO-presentation-of-report1*, in which a global element *view-property* is used to view the initial report1. In fact the rotation of artifacts is implemented through rotationally binding environments with activities in the original design.

```
<learning-activity identifier="LA-write-initial-report1">
    <title>Write Report1</title>
    <environment-ref ref="ENV-for-report1"/>
    <activity-description>
        <title>Write Report</title>
        <item identifier="ITEM-write-report1" identifierref="RESO-write-
report1" />
    </activity-description>
</learning-activity>
```

Figure 78a: the definition of the activity, in which learner1 creates the initial report1

```
<p>Please write the initial report.</p>
<set-property ref="InitialReport1" property-of="self" />
```

Figure 78b: the content of the resource file "RESO-write-report1"

```
<learning-activity identifier="LA-comment-2-1">
    <title>Learner2 comments on report1</title>
    <environment-ref ref="ENV-for-report1"/>
    <activity-description>
        <title>Commenting</title>
        <item identifier="ITEM-write-comment-2-1" identifierref="RESO-
comment-2-1" />
    </activity-description>
</learning-activity>
```

Figure 78c: the definition of the activity that is associated with an environment

```
<environment identifier="ENV-for-report1">
    <title>working environment for report1</title>
    <learning-object identifier="LO-information-about-report1">
        ……
        <item identifier="ITEM-report1" identifierref="RESO-presentation-
of-report1" isvisible="false">
            <title>report1</title>
        </item>
        ......
    </learning-object>
</environment>
```

Figure 78d: the definition of the environment storing the initial report1

```
<h3>Initial Report 1:</h3>
<view-property ref="InitialReport1" view="value"/>
```

Figure 78e: the content of the resource file "RESO-presentation-of-report1"
Figure 78. Transference of an artifact via an environment

Another solution is to present all imported artifacts in the same information item of the activity which consumes the artifacts. Rather than using an environment, the artifact is transferred by means of the activity-descriptions of the activities which produce and consume the artifact. Because of the limited space available here we omit the code illustrating this approach.

Usability: as mentioned above, in IMS-LD a property can be used to represent artifacts. Because a property in IMS-LD has a primitive data type such as *integer*, *string*, *duration*, etc. the coordination mechanism for managing usability dependency is simply to check the data type and constraints of the property. In this use case, all properties should be defined as type *text*.

Representation of Coordination Mechanisms to Manage Sharing Dependencies

In IMS-LD task allocation is represented as a role-part. As shown in Figure 4, a set of role-parts are defined to represent three learners who are assigned to perform different activities. These activities share the labor resources at different times.

We can represent another coordination mechanism for managing sharing dependencies in IMS-LD: three sequential activities in each of which three learners work together. Each activity is designed as a collaborative activity leading to the production of a report. Each activity has an environment containing certain learning services such as chat, forum, shared text editor, shared whiteboard, audio/video conferencing, and so on. As mentioned before, in a fluid collaboration learners can use these collaborative tools to coordinate their actions at a finer-grained level and produce shared artifacts. Because the code representing this coordination mechanism is extensive it is not included here.

IMS-LD provides static coordination mechanisms for managing sharing dependencies, but it is difficult to support dynamic coordination mechanisms, for example, the "first come, first served" mechanism. We can investigate how to model an alternative design, in which tasks are assigned to roles according to the time sequence that users register to the execution. Using this approach it is unpredictable at design time who will come first in an actual execution, unlike a pre-defined allocation of tasks as role-parts described in Figure 75. Because the XML code to implement this mechanism is too extensive we describe and explain it using pseudo-code as shown in Figure 79.

In order to control the execution of activities at the right time, data-driven mechanisms (similar to the code shown in Figure 77) are needed as a complete coordination mechanism. Figure 79a declares three roles: *learner1*, *learner2*, and *learner3* and sixteen activities: three registering and twelve activities illustrated in Figure 72. Figure 79b declares three activity-structures and each activity-structure consists of four sequential activities: writing the initial report, commenting on the reports of two peers, and creating the final report. Figure 79c defines three properties representing the time when learners finish the registration. Figure 79d specifies how the values of three properties are assigned. Because three learners may complete registration at different points of time, the *current time* assigned by the system will have different values for different learners. In Figure 79e, the first statement specifies that if *learner1* and *learner2* have registered and *learner3* has not finished registration, and *learner1* registered before *learner2* did (or they registered at the same time) then *learner1* will be assigned to perform *activity-structure1*, *learner2* will be responsible for doing *activity-structure2*, and *activity-structure3* will be carried out by *learner3*. The following five statements specify the allocation tasks in the other five situations, in which three learners finish the registrations in different time sequences.

```
<!— the three roles and twelve activities are defined as those defined in
the original design. -->
Role: learner1, learner2, learner3;
Activity: registering1, registering2, registering3, reporting1, ......,
revising3;
```

Figure 79a: the declaration of three roles and twelve activities

```
<!— the four activities performed by the same learner are defined as a
sequence activity-structure. Therefore, three activity-structures are
defined -->
Activity-structure: activity-structure1 := reporting1 + criticizing1-3 +
criticizing1-2 + revising1;
                    activity-structure2 := reporting2 + criticizing2-1 +
criticizing2-3 + revising2;
                    activity-structure3 := reporting3 + criticizing3-2 +
criticizing3-1 + revising3;
```

Figure 79b: three properties are defined for representing when each learner registers

```
<!— three properties are defined for representing when each learner
registers -->
Property: T1 := 0, T2 := 0, T3 := 0;
```

Figure 79c: when a learner has finished registration, the registration time will be recorded

```
<!— when a learner has finished registration, the registration time will be recorded -->
If (registering1 complete) then T1 := current time;
If (registering2 complete) then T2 := current time;
If (registering3 complete) then T3 := current time;
```

Figure 79d: according to the sequence in which three learners register, the activity structures will be assign to the learners in the way first-come-first-served

```
<!— according to the sequence in which three learners register, the activity structures will be assign to the learners in the way first-come-first-served -->
If ((T1 is not 0) and (T2 is not 0) and (T3 is 0) and (T1<=T2)) then
notification (learner1 activity-structure1), notification (learner2
activity-structure2), notification (learner3 activity-structure3);

If ((T1 is not 0) and (T2 is not 0) and (T3 is 0) and (T1>T2)) then
notification (learner1 activity-structure2), notification (learner2
activity-structure1), notification (learner3 activity-structure3);

If ((T1 is not 0) and (T2 is 0) and (T3 is not 0) and (T1<=T3)) then
notification (learner1 activity-structure1), notification (learner2
activity-structure3), notification (learner3 activity-structure2);

If ((T1 is not 0) and (T2 is 0) and (T3 is not 0) and (T1>T3)) then
notification (learner1 activity-structure2), notification (learner2
activity-structure3), notification (learner3 activity-structure1);

If ((T1 is 0) and (T2 is not 0) and (T3 is not 0) and (T2<=T3)) then
notification (learner1 activity-structure3), notification (learner2
activity-structure1), notification (learner3 activity-structure2);

If ((T1 is 0) and (T2 is not 0) and (T3 is not 0) and (T2>T3)) then
notification (learner1 activity-structure3), notification (learner2
activity-structure2), notification (learner3 activity-structure1);
```

Figure 79e: notifications are used to allocate tasks dynamically
Figure 79. An example of dynamic coordination mechanism

If notification is not used, it is necessary to enumerate all possible role-parts in the same act (the total number of turples is the combination of the number of roles and the number of activities, 3*12=36 in this use case), and set them to *invisible*. After the rotation pattern is determined, 12 activities are set to *visible* to make 12 associated role-parts active. If the number of users and cases increases, the complexities of the process model increase accordingly. The difficulties in representing dynamic coordination mechanisms are ascribed to a) no *identifier* data type and no *collection* data type specified for the property and b) insufficient operations such as 'find a person whose personal property meets a condition', 'add a person as an active role', 'add a role-part within an act', and so on.

9.3.4. Future trends: the reuse of coordination mechanisms

As we have seen, representing coordination mechanisms is a time-consuming and error-prone task. It is necessary to explore whether coordination mechanisms can be represented at a more abstract level than XML, that is to say at a higher level than the executable code. It is expected that the abstract representation could be more intuitively understood and used by practitioners (e.g., instruction designers and teachers) who do not have sophisticated technical knowledge and skills. The system would then automatically transform such an abstract representation into XML code. This process provides a means whereby coordination mechanisms could be reused without requiring users to understand how the executable code works. In this section we discuss issues related to such reuse.

9.3.4.a. Identifying common dependencies and the mechanisms for managing them

According to coordination theory, dependencies and the mechanisms for managing them are *general*, which means that a given dependency and a mechanism to manage it will be found in a variety of settings. For example, a common coordination problem appears when certain activities require specialized competences, thus constraining which persons can work on them. This kind of dependency arises in many situations and there is a generic set of coordination mechanisms (managing this dependency) which appear over and over in different processes. Coordination theory also describes how several coordination mechanisms can often be used to manage a dependency. For example, mechanisms to manage sharing a dependency between roles and activities can include qualification-checking, priority-comparing, first-come-first-served, and so on. Because of this it is valuable to identify and study common dependencies and their related coordination mechanisms, in order to facilitate reuse.

9.3.4.b. Reusing computational coordination mechanisms

Once the dependencies and corresponding coordination mechanisms have been identified, the next step is to represent the coordination mechanisms in IMS-LD. As we have seen, the representation of some coordination mechanisms in IMS-LD is a very complex task, even for users with sound technical knowledge. It is therefore desirable to make the representation of coordination mechanisms reusable. Through an analysis of the IMS-LD manifest file and resource files, we have found that some parts of code are static and some parts of code are replaceable and related to particular elements. We can therefore store a fragment of code as an executable component in a library of an IMS-LD authoring environment. We can refer to this using an abstract representation, which can have parameters with values which are assigned by the user in design-time. For example, if a user wants to model the transference of a document from one activity to another, s/he can use an abstract representation: transfer a document (parameter1) from an activity (parameter2) to another activity (parameter3). The constraints for the parameters are that parameter1 must be a property reference representing a document to be transferred, parameter2 and parameter3 must be activity references. Once the user has applied a coordination mechanism (by choosing the corresponding abstract representation and assigning the values to parameters) the system automatically maps the abstract representation to the component.

In the same way, more complex coordination mechanisms needed in the 'Knowledge Convergence Script' can be represented as well. For instance, the abstract representation: distribute documents (document1, document2, document3) within activities (activity1, activity2, activity3) indicates the one-to-one distribution of three documents between three activities. Similarly the abstract representation: rotate documents (document1, document2, document3) from activities (activity1, activity2, activity3) to succeeding activities (activity4, activity5, activity6) means to transfer three documents produced in three activities to three succeeding activities as follows:

- Transferring document1 produced in activity1 to activity5
- Transferring document2 produced in activity2 to activity6
- Transferring document3 produced in activity3 to activity4

It is clear that a high-level representation of coordination mechanisms of this kind is much easier to understand and use than a concrete representation codified using IMS-LD and expressed in XML (see Figure 76), or using a programming language (e.g., JAVA). Currently, we are working on developing a high-level modeling language and mapping algorithms to transform a group-based learning design represented in the high-level modeling language to an executable model represented in IMS-LD. This work is technical in nature, and so we do not discuss the details in this chapter.

9.3.5. Conclusions

This research is a theory-based analysis. First, we briefly introduce group-based learning and coordination theory. Using coordination theory as an analytical framework we analyze dependencies and possible coordination mechanisms for managing them in group-based learning. We identify a variety of dependencies and some related coordination mechanisms through the investigation of a use case and some of its variants. We then analyze the expressiveness of IMS-LD in representing the identified coordination mechanisms. We conclude that in supporting group interaction it is possible to represent almost all basic coordination mechanisms in IMS-LD. In particular, IMS-LD provides sufficient mechanisms to manage: task and role decomposition dependencies, weak and strong precedence dependencies, and static resource sharing dependencies. However, we have also recognized that the representation of certain coordination mechanisms presents some challenges. Specifically it is complex to represent: the coordination of the assembly of components, transference of artifacts in some complicated distribution patterns, complicated group formation and group dynamics, and allocation of tasks and resources using some dynamic coordination mechanisms. The reasons for these difficulties are briefly analyzed and possible solutions are also discussed.

Based on this analysis, we have briefly explored the feasibility of reusing coordination mechanisms in modeling group-based learning processes. In comparison with IMS-LD code in the form of XML, a representation of common coordination mechanisms at a high-level of abstraction may be more intuitively understood and used by practitioners. We are currently identifying and codifying generic coordination mechanisms which will be archived as a library in the IMS-LD authoring environment for reuse on future occasions. We will implement an advanced IMS-LD authoring environment in which the user can design group-based learning processes using the abstract representation. The system will then automatically generate IMS-LD code based on abstract representations and the executable components in the library.

9.3.6. Acknowledgements

The work on this paper has been sponsored by the TENCompetence Integrated Project that is funded by the European Commission's 6th Framework Programme, priority IST/Technology Enhanced Learning (Contract 027087).

10. Annex II: case studies and Units of Learning

10.1 . Caso de estudio Nivel B: Global Warming

El siguiente ejemplo, Global Warming, muestra algunas de las características principales del Nivel B. En concreto: Multiple roles, Collaboration, A learning service, Properties, Completion rules, Conditions, Showing and Hiding and Global elements. Representa pues una buena muestra de las posibilidades del Nivel B con elementos de base del Nivel A. Este ejemplo ha sido construido sobre la base de `What is Greatness`, desarrollado en una primera fase por Colin Tattersall y en una segunda fase por él mismo y por el autor. El diagrama UML que muestra el flujo de información es el siguiente (Figure 80):

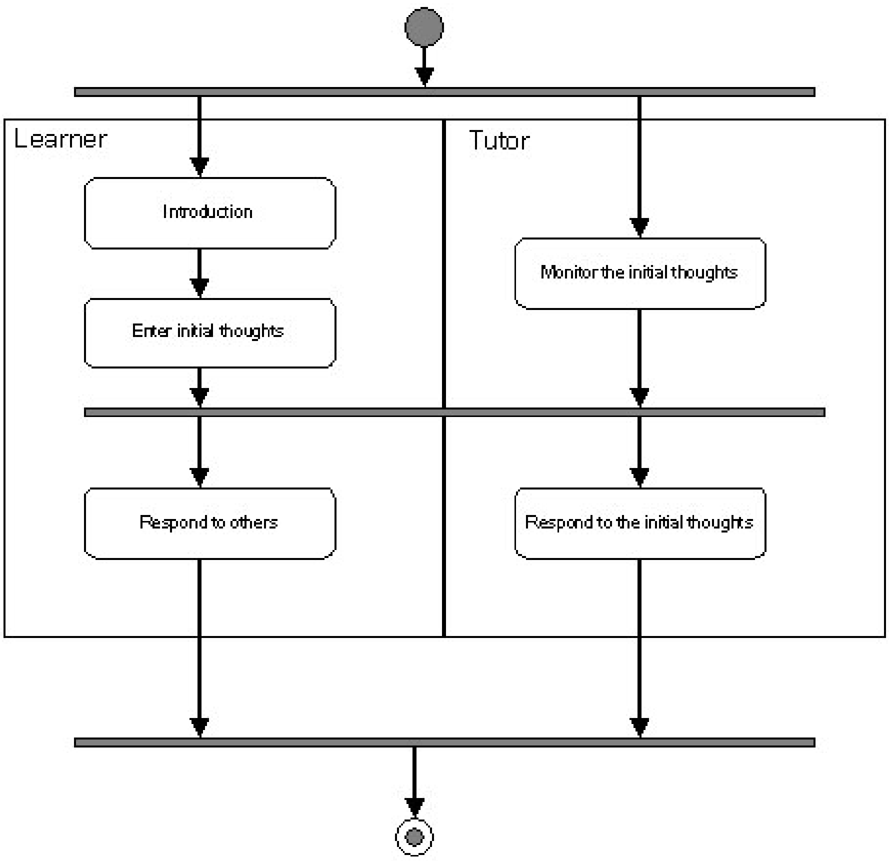


Figure 80. Diagrama UML mostrando Itinerarios y roles en la UoL Global Warming

La secuencia del proceso hace que en un primer momento el estudiante lea una introducción sobre el problema del calentamiento global del planeta y sobre la Cumbre de Kyoto e introduzca sus primeras impresiones en un formulario. Mientras, el profesor monitoriza este proceso, en concreto las impresiones. Cuando lo considera oportuno, el profesor cierra esta etapa y sincroniza los dos roles. En ese momento, los estudiantes pueden ver las impresiones de los demás y hacer comentarios. El profesor también puede ver todas las impresiones y los comentarios posteriores, proporcionando una reflexión final y cerrando el proceso cuando considera oportuno.

En esta UoL las dependencias y el flujo de información se realizan no únicamente en el manifiesto (como ocurría en el ejemplo Candidas sobre el Nivel A) sino en diversos ficheros XHTML adicionales. La siguiente tabla (Table 22) muestra la estructura general de estos ficheros y su función:

Table 22. Estructura general de ficheros y funciones en "Global Warming"

Fichero	Función
imsmanifest.xml	Definición general del learning design, incluyendo el método, los roles, las actividades de aprendizaje y de soporte, los entornos y las condiciones
initial-thoughts-form.xml	Formulario donde introducir los comentarios iniciales de cada usuario. Contiene código HTML y Global Elements
initial-thoughts-overview.xml	Informe donde se pueden ver los comentarios iniciales de los demás miembros. Contiene Global Elements que estarán enlazados con un Entorno de una Actividad originando un servicio de monitorización
learning-objectives.xml	Descripción de los objetivos de aprendizaje. También podría ser HTML en este caso concreto
prerequisites.xml	Descripción de los prerrequisitos. También podría ser HTML en este caso concreto
response-to-initial-thoughts-form.xml	Comentarios sobre las aportaciones de los demás. También permite modificar las aportaciones propias. Contiene HTML, Global Elements (view-property y set-property) y capas DIV para que el método en el manifiesto pueda ocultar y mostrar información
set-activity2-complete.xml	Muestra y oculta información en capas DIV y permite completar cerrar la segunda actividad con un set-property
initial-thoughts-tutor-form.xml	El tutor visualiza los comentarios de los estudiantes y puede proporcionar a su vez una respuesta

	personalizada. Además, aquí puede cerrar esta tercera actividad, sincronizando a todos los usuarios y dando por concluida la unidad de aprendizaje

Las características específicas de Nivel B en cada uno de los ficheros son:

Fichero: initial-thoughts-form.xml

```
<div class="C-Activity2-not-complete">
    <p>Please indicate whether you are for or against the proposed
Global Warming official response in the Kyoto Summit.</p>
    <set-property ref="LP-Opinion"/>
    <p>Give some rationale for your opinion. Subsequently, these
thoughts will be made available to the whole group, and you will be able
to see the thoughts of others.</p>
    <set-property ref="LP-initial-thoughts"/>
</div>

<div class="C-Activity2-complete">
    <p>You stated that you were <I><ld:view-property ref="LP-Opinion"
property-of="self" view="value"/> </I> the proposed Global Warming.</p>
    <p>Your rationale was</p>
    <q>
        <view-property ref="LP-initial-thoughts" property-of="self"
view="value"/>
    </q>
    <p>The activity has now been completed.</p>
</div>
```

Nota: Se introducen y se visualizan los comentarios de uno mismo (`self`) en dos propiedades `LP-Opinion` y `LP-initial-thoughts`, utilizando dos global elements view-property que se introducen en dos capas DIV para poder ser mostradas u ocultadas, según la actividad está abierta o cerrada, tal y como se define en el manifiesto

Fichero: initial-thoughts-overview.xml

```
<P>This learner is <I>< view-property property-of="supported-person"
ref="LP-Opinion"/></I> the proposed topic and supplied the following
rationale:</P>
<view-property property-of="supported-person" ref="LP-initial-
thoughts"/>
```

Nota: Se visualizan los comentarios de los demás (supported-person). Esta característica debe configurarse como un servicio de monitorización dentro de un entorno descrito en el manifiesto y asociado a una actividad

Fichero: response-to-initial-thoughts-form.xml

```
<div class="C-UoL-not-complete">
         <p>You are <I><ld:set-property ref="LP-Opinion"/></I> the
proposed Global Warming topic. Your rationale being:</p>
         <view-property ref="LP-initial-thoughts" property-of="self"
view="value"/>
         <p>Take a look at what others think… Go to the environment</p>
         <set-property ref="LP-response-to-initial-thoughts"/>
</div>
<div class="C-UoL-complete">
     <P>Your final position was to be <I><ld:view-property ref="LP-
Opinion"/></I> the proposed Global Warming topic.</P>
</div>
```

Nota: En dos capas DIV (not-complete y complete) se muestra y solicita diferente información mediate set-property y view-property. Se propone acudir al entorno asociado, donde se encuentra definido un servicio de monitorización que ejecutará el fichero initial-thoughts-overview.xml

Fichero: set-activity2-complete.xml

```
<div class="C-Activity2-not-complete">
   <p>Complete the activity by setting this variable to true:</p>
   <set-property ref="LP-activity-2-completed"/>
</div>
```

Nota: El tutor puede cerrar la segunda actividad mediante la configuración con un set-property de la propiedad LP-activity-2-completed

Fichero: initial-thoughts-tutor-form.xml

```
<p>Initial thoughts:</p>
<view-property ref="LP-initial-thoughts" property-of="supported-person"
view="value"/>
<p>Response to others:</p>
<view-property ref="LP-response-to-initial-thoughts" property-
of="supported-person" view="value"/>
<p>You can enter comments on the initial thoughts here:</p>
<set-property ref="LP-tutor-comments-initial-thoughts" property-
of="supported-person"/>
```

```
<p>You can set the activity to completed for this learner here:</p>
<set-property ref="LP-activity-3-completed" property-of="supported-
person"/>
```

Nota: El tutor puede ver comentarios de otros y respuestas a los comentarios de otros. También puede responder a cada usuario de manera personal (supported-person) y cerrar la actividad para cada usuario mediante un set-property. Una vez que el tutor ha cerrado la actividad a todos el flujo continúa con la siguiente actividad

Fichero: imsmanifest.xml

```
<learning-activity identifier="LA-enter-initial-thoughts"
isvisible="true">
    <title>Enter your initial thoughts</title>
    <environment-ref ref="E-general-environment" />
    <activity-description>
        <item identifier="I-enter-initial-thoughts"
identifierref="resource-1 " isvisible="true">
        <title>Note down your thoughts on the proposed topic about
Global Warming</title>
        </item>
    </activity-description>
            <complete-activity>
              <when-property-value-is-set>
                <property-ref ref="LP-activity-2-completed" />
                <property-value>true</property-value>
              </when-property-value-is-set>
            </complete-activity>
</learning-activity>
```

```
<resource identifier="resource-1" type="imsldcontent" href="initial-
thoughts-form.xml">
      <file href="initial-thoughts-form.xml" />
      <file href="globalwarming.css" />
</resource>
```

Nota: Esta actividad inicial enlaza con la Activity-Description que se encuentra en el resource-1 y que corresponde con el fichero initial-thoughts-form.xml. A su vez, esta actividad se completa cuando la propiedad LP-activity-2-completed is true, algo que ocurre cuando el tutor la activa mediante el fichero set-activity2-complete.xml.

```
<learning-activity identifier="LA-respond-to-others" isvisible="false">
    <title>Respond to the thoughts of others</title>
```

```
    <environment-ref ref="env-2" />
    <activity-description>
              <item identifier="I-respond-to-others-1"
identifierref="resource-2a" isvisible="true">
                <title>Enter your response</title>
              </item>
    </activity-description>
</learning-activity>
```

```
          <environment identifier="env-2">
            <title>What did other learners say?</title>
            <service identifier="service-2" isvisible="true">
              <monitor>
                <role-ref ref="Learner" />
                <title>Others' thoughts</title>
                <item identifier="item-2" identifierref="resource-2b"
isvisible="true" />
              </monitor>
            </service>
          </environment>
```

```
<resource identifier="resource-2b" type="imsldcontent" href="initial-
thoughts-overview.xml">
      <file href="initial-thoughts-overview.xml" />
</resource>
```

Nota: Esta actividad permite responder a los comentarios de otros estudiantes. Para ello enlaza el entorno `env-2`, que a su vez define un servicio de monitorización `service-2` que enlaza con el recurso `resource-2b`, que no es más que el fichero de tipo `imsldcontent` `initial-thoughts-overview.xml`, descrito con anterioridad. Este sistema de referencias en cascada actividad-entorno-servicio-recurso-fichero es habitual en cualquier especificación eLearning y, en general, en cualquier especificación sobre gestión de conocimiento o información.

```
<act>
…
<complete-act>
    <when-role-part-completed ref="RP-Facilitator-1" />
</complete-act>
</act>
<complete-play>
```

```
    <when-last-act-completed />
</complete-play>
</play>
<complete-unit-of-learning>
    <when-play-completed ref="P-1" />
</complete-unit-of-learning>
```

Nota: Diversas condiciones de terminación para un act, un play y la UoL

```
<if>
    <complete>
        <learning-activity-ref ref="LA-enter-initial-thoughts" />
    </complete>
</if>
<then>
    <show>
        <class with-control="false" class="C-Activity2-complete" />
    </show>
    <hide>
        <class with-control="false" class="C-Activity2-not-complete" />
    </hide>
</then>
    <else>
      <show>
         <class with-control="false" class="C-Activity2-not-complete" />
      </show>
      <hide>
         <class with-control="false" class="C-Activity2-complete" />
      </hide>
    </else>
```

Nota: Cuando la actividad `LA-enter-initial-thoughs` está completada (algo que indica el tutor a través del fichero `initial-thoughts-tutor-form.xml`) se muestra la capa `C-Activity2-complete` y se oculta la capa `C-Activity2-not-complete` que se encuentran en los ficheros `initial-thoughts-form.xml` y `set-activity2-complete.xml`. Es decir, con una misma condición actuamos en dos ficheros distintos al mismo tiempo

10.2 . **Caso de estudio Nivel C: Ampliación del Nivel B**

Al ejemplo anterior, y como extensión del Nivel C, se podría añadir una notificación a la terminación de una de las actividades que permitiera enviar un correo electrónico al tutor de manera automática. Para ello, hacen falta tres cosas: Una `global property` con el correo electrónico, una definición del mensaje a enviar cuando se complete la actividad y una configuración del sistema SMTP de envío de correo

```
<globpers-property identifier="prop-email">
   <global-definition uri="http://whoknows.nl">
         <title>Email address</title>
         <datatype datatype="string" />
         <initial-value>daniel.burgos@ou.nl</initial-value>
   </global-definition>
```

Nota: Incluye la dirección de correo electrónico de envío como valor por defecto

```
<on-completion>
   <notification>
         <email-data email-property-ref="prop-email">
               <role-ref ref="role-student" />
         </email-data>
         <subject>Cheers David</subject>
   </notification>
</on-completion>
```

Nota: En una condición de terminación de una `learning activity` y asociado a un rol determinado y a una propiedad que contiene la dirección de correo electrónico se incluye un mensaje

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--  $Id: mail-service.xml,v 1.4.2.2 2003/10/13 starksm Exp $ -->
<server>
<mbean code="org.jboss.mail.MailService" name="jboss:service=Mail">
  <attribute name="JNDIName">java:/Mail</attribute>
  <attribute name="User">login</attribute>
  <attribute name="Password">password</attribute>
  <attribute name="Configuration">
  <configuration>
    <property name="mail.store.protocol" value="pop3" />
    <property name="mail.transport.protocol" value="smtp" />
    <property name="mail.user" value="nobody1@nohost.nodomain.com" />
    <property name="mail.pop3.host" value="pop3.nohost.nodomain.com" />
    <property name="mail.smtp.host" value="smtp.nohost.nodomain.com" />
    <property name="mail.from" value="nobody2@nohost.nodomain.com" />
    <property name="mail.debug" value="false" />
  </configuration>
  </attribute>
```

```
</mbean>
</server>
```

Nota: Se configuran todos los parámetros de un servicio estándar de correo electrónico POP3/SMTP, aunque únicamente el SMTP es necesario para la ejecución de las notificaciones. Esta personalización es dependiente del motor de ejecución. En este ejemplo concreto hemos utilizado la configuración del motor Coppercore [3], modificando el fichero `mail-service.xml`.

10.3 . Case study, adaptation on the learner´s performance and knowledge: GeoQuiz

Note about authoring: every case study that we show from now onwards has been modelled using the simple text editor NotePad and the XML editor XMLSpy, and they have been published and executed with CopperCore 3.0. Afterwards, all of them have been revisited using several IMS-LD Editors, like CopperAuthor, Cosmos, Reload IMS-LD Editor and ASK LDT, although none of them provide with a friendly, high-level interface that could help on modeling. From a low-level perspective, Cosmos is the most effective, stable and intuitive one. Although it is out of the scope of this dissertation to evaluate these editing-authoring tools, it is also a fact that tooling is one of the keys for a further implementation of IMS-LD with actual teachers and students. Thus, a few remarks based on the case studies could be informally related to the tools on ocassion.

The Unit of Learning GeoQuiz, along three versions I, II y III, but even more in III, is focused on adaptation based on the performance and the previous knowledge of the learner (Figure 81). The learning designer has previously to define the set of rules that manage the learner´s status. The flow basically adapts the next action to take after a quiz is fulfilled. Depending on the answers and on the threshold to achieve, one or another activity is turned on, what actually means that the learner reaches a different, adapted level. The learning designer defines a set of questions and answers and three possible outcomes, meaning one activity per outcome. This learning flow is shown in Figure 82. Although firstly hidden, after the questionnaire one out of the three possible activities is shown. No interaction with a teacher is established. We will use this case study as a base to build upon. The rest of the case studies will show punctual contributions to this very detailed example. Following, we show the key parts of the code.

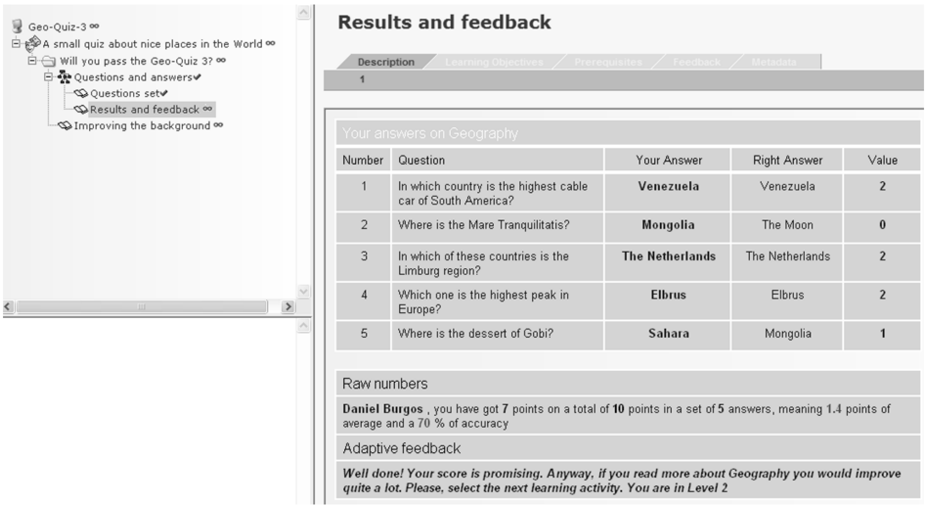


Figure 81. GeoQuiz III

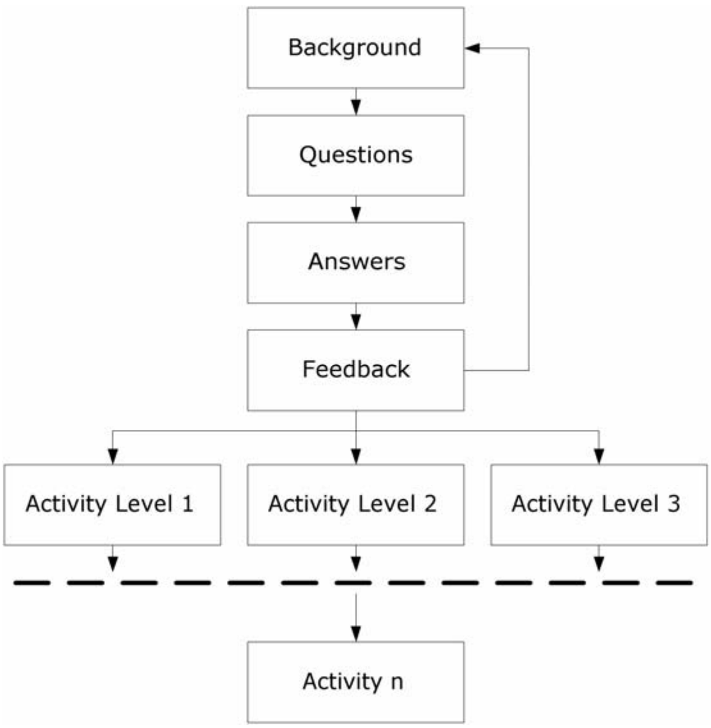


Figure 82. Learning flow in GeoQuiz

Before the actual start of the quiz, the name of the user is requested and stored in a property called Name, using the global element <set-property> in an external XML file:

```
Please, enter your name: <set-property
xmlns="http://www.imsglobal.org/xsd/IMS-LD_v1p0" ref="Name" property-
of="self" view="value"/>
```

Afterwards, the questions are raised, from another external XML file that shows the properties with the questions and the possible answers, **Answer1**:

```
  <tr>
    <td height="35">Question 1</td>
```

```
    <td>In which country is the highest cable car of South America?</td>
    <td><p><set-property xmlns="http://www.imsglobal.org/xsd/IMS-
LD_v1p0" ref="Answer1" property-of="self" view="value"/></p></td>
  </tr>
```

Once all the questions are answered, a DIV layer **Answered** shows up with a supporting message:

```
<div class="Answered">
   <p>Perfect. You have entered your answers. Do you want to check your
results? Please, refresh your browser and go ahead</p>
</div>
```

The results are shown now, with <**view-property**>, using a new XML file with all the properties involved in this feedback step. On the one hand, the right answers to the questions, **Value1**:

```
  <tr>
    <td>1</td>
    <td>In which country is the highest cable car of South America?
</td>
    <td><p><view-property ref="Answer1" property-of="self"
view="value"/></p></td>
    <td>Venezuela</td>
    <td><p><view-property ref="Value1" property-of="self"
view="value"/></p></td>
  </tr>
```

On the other hand, the interpretation of the results, showing the properties **Name**, **Total**, **Average** and **Accuracy**, coming up after several calculations:

```
<view-property ref="Name" property-of="self" view="value"/>,
        you have got <view-property ref="total" property-of="self"
view="value"/>points
        on a total of 10 points in a set of 5 answers,
        meaning <view-property ref="average" property-of="self"
view="value"/> points of average and a <view-property ref="accuracy"
property-of="self" view="value"/>% of accuracy
```

Also, the property with the next action to take is shown, **prop-feedback**:

```
<view-property ref="prop-feedback" property-of="self" view="value"/>
```

And the new related Learning Activity is visible in the player (meaning any kind of resource, an HTML file, for instance).

All these steps in the learning flow are possible because of the previous setting in the manifest file, a) setting properties:

```
<locpers-property identifier="Name">
   <datatype datatype="string"/>
</locpers-property>
```

```
<locpers-property identifier="prop-feedback">
   <datatype datatype="string"/>
</locpers-property>
```

```
<locpers-property identifier="Answer1">
   <datatype datatype="string"/>
   <initial-value>Select</initial-value>
   <restriction restriction-type="enumeration">Select</restriction>
   <restriction restriction-type="enumeration">Venezuela</restriction>
   <restriction restriction-type="enumeration">Peru</restriction>
   <restriction restriction-type="enumeration">Ecuador</restriction>
</locpers-property>
```

b) Defining the rule for completion of the learning activity with the set of questions. This rule is the key to provide an adaptive next step to take, as the next activity cannot be shown until this learning activity is closed:

```
<learning-activity identifier="questions">
   <title>Questions set</title>
   <learning-objectives>
   <item identifierref="res-objectives" identifier="I2-objectives">
         <title>Learning objectives</title>
   </item>
</learning-objectives>
<activity-description>
   <item identifierref="res-questions" identifier="I-questions" />
</activity-description>

<complete-activity>
   <when-property-value-is-set>
         <property-ref ref="all-questions"/>
         <property-value>1</property-value>
   </when-property-value-is-set>
</complete-activity>
</learning-activity>
```

c) Defining rules to check conditions. In the following case, Value1 is assigned with 1 or 2 depending on the content of the property Answer1:

```
<if>
   <is>
         <property-ref ref="Answer1"/>
         <property-value>Venezuela</property-value>
   </is>
</if>
<then>
   <change-property-value>
         <property-ref ref="Value1"/>
         <property-value>2</property-value>
   </change-property-value>
</then>
<else>
   <if>
         <is>
               <property-ref ref="Answer1"/>
               <property-value>Peru</property-value>
         </is>
   </if>
   <then>
         <change-property-value>
               <property-ref ref="Value1"/>
               <property-value>1</property-value>
         </change-property-value>
   </then>
</else>
```

d) Defining and hiding DIV classes, like **Answered**, that is shown after all the questions are fulfilled (true=1):

```
<if>
   <is>
         <property-ref ref="all-questions"/>
         <property-value>1</property-value>
   </is>
</if>
<then>
   <show>
         <class class="Answered" />
```

```
   </show>
</then>
   <else>
         <hide>
         <class class="Answered" />
   </hide>
   </else>
```

e) Making calculations, a division between **total** and **number-questions** to get the **average**:

```
<change-property-value>
   <property-ref ref="average" />
   <property-value>
         <calculate>
               <divide>
                     <property-ref ref="total" />
                     <property-ref ref="number-questions" />
               </divide>
         </calculate>
   </property-value>
</change-property-value>
```

And f) defining the key for the adaptive feedback and the next action to take. Depending on the accuracy only one activity is shown, `flow1` (0-49), `flow2` (50-75) or `flow3` (76-100), together with a message of feedback in prop-feedback. These activities are linked to HTML resources:

```
<if>
   <less-than>
         <property-ref ref="accuracy"/>
         <property-value>1</property-value>
   </less-than>
</if>
<then>
   <hide>
         <learning-activity-ref ref="flow1" />
         <learning-activity-ref ref="flow2" />
         <learning-activity-ref ref="flow3" />
</hide>
</then>
<else>
<if>
```

```
   <and>
         <greater-than>
               <property-ref ref="accuracy"/>
               <property-value>0</property-value>
         </greater-than>
         <less-than>
               <property-ref ref="accuracy"/>
               <property-value>50</property-value>
         </less-than>
   </and>
</if>
<then>
   <change-property-value>
         <property-ref ref="prop-feedback"/>
         <property-value>Insuficient.You are in Level 1</property-value>
   </change-property-value>
   <show>
         <learning-activity-ref ref="flow1" />
   </show>
   <hide>
         <learning-activity-ref ref="flow2" />
         <learning-activity-ref ref="flow3" />
   </hide>
</then>
<else>
<if>
   <and>
         <greater-than>
               <property-ref ref="accuracy"/>
               <property-value>49</property-value>
         </greater-than>
         <less-than>
               <property-ref ref="accuracy"/>
               <property-value>76</property-value>
         </less-than>
   </and>
</if>
<then>
   <change-property-value>
         <property-ref ref="prop-feedback"/>
         <property-value>Well done! Level 2</property-value>
```

```
   </change-property-value>
   <show>
         <learning-activity-ref ref="flow2" />
   </show>
   <hide>
         <learning-activity-ref ref="flow1" />
         <learning-activity-ref ref="flow3" />
   </hide>
</then>
<else>
   <change-property-value>
         <property-ref ref="prop-feedback"/>
         <property-value>Congratulations! Level 3</property-value>
   </change-property-value>
   <show>
         <learning-activity-ref ref="flow3" />
   </show>
   <hide>
         <learning-activity-ref ref="flow1" />
         <learning-activity-ref ref="flow2" />
   </hide>
</else>
</else>
</else>
```

In the next examples we will only show the key aspects in every case study, in order to make a neater coding and to focus on the real essencial contributions to the topic on adaptive eLearning.

10.3.1. **Remarks on this case study**

- The definition of properties and the link through several working XML files is too complicated to become useful. Even a simple combobox comes to a struggle that has to be defined by hand, as the current editing level of the available editors is too low or too messy
- The relation between layers and actions is not straightforward and it has to be done interlacing files, once more
- The lack of a richer conditional structure makes the editing of the set of rules more complicated on paper than they actually are from a logical point of view

- In case that a user fails and wants to re-fill the questionnaire, how can he/she make it? Once that the questions are answered and the activity is closed (the on completion flag is switched on) the activity cannot be re-initialized and go backwards. Iterations in the activities are not allowed, therefore
- In case that the teacher or the learning designer wants to change the questions, or the answers, or the content of the next activity to carry out, it is not possible to do it, as every single resource has to be packed in design and publishing time before the actual running of the instance. Changes on-the-fly are not possible

10.4 . Case study, adaptation on the learning designer´s method: QuoBuilder II

Changing things on-the-fly becomes a tricky issue with IMS-LD. The specification allows for it, but the current engines (CopperCore, netUniversité, .LRN…) don´t. Therefore, we have to deal with a limitation of the tools, and not of the specification itself. However, in the current state of the art, we don´t have the chance to change it, as no other engine is available and all of them are based on the phases of validation and publication.

Nevertheless, a learning designer is able to modify things on the run, as long as they are planned before. We show it in Section 4. In the following Unit of Learning (Figure 83), a whole questionnaire is created empty in design-time and it is adapted by the teacher/learning designer in the run-time, with no further need of re-compilation.

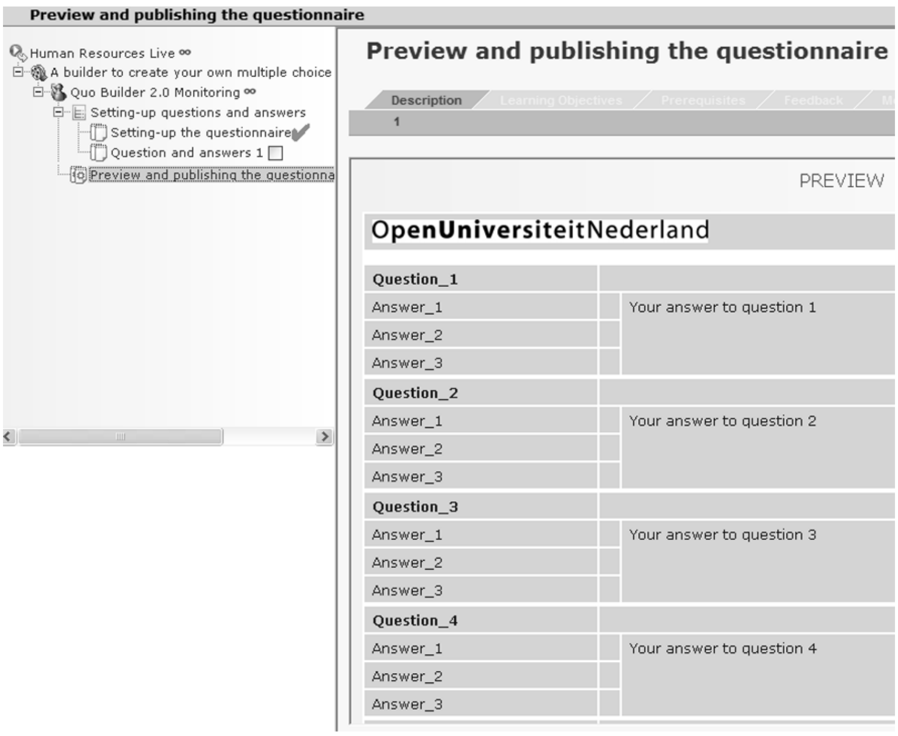


Figure 83. Quo Builder II

The main features are:

- Setting-up of the Unit of Learning with full personalisation of questions, answers, right answers, ranges, points earned, messages of feedback and welcome, title
- Questions and related properties are local (loc-property) and keep the same value for all the users in the same run but personal answers and calculations are private and linked to every participant (locpers-property)
- Five questions with three possible answers. If the answer is right earns the ammount of points defined in the set-up
- Only when you have answered the five questions you can go ahead to see the results (don´t forget to refresh your browser)
- A total sum, a simple average and a porcentage of accuracy are calculated
- An adaptive feedback is provided depending on the accuracy (there are four ranges, also defined in the set-up)
- The next activity delivered depends also on the feedback. If the student doesn't reach the lowest threshold there is no next activity and he/she has to repeat the questionnaire
- There are 2 roles, teacher and participant, and the learning flow swaps between them
    - First, the teacher sets-up the questionnaire and the particiant waits for the opening of the course. The teacher can have a preview of the questionnaire before publishing
    - Second, the teacher publishes the quo and the participant answers the questions. The teacher monitors his/her progress
    - Third, the participant finishes the quo and receive two inputs: an adaptive feedback and a new activity, both based on the results
- The logo and the next activities (Level 1, 2 and 3) can be easily changed in the ZIP file to fit them to personal goals of the teacher in an easy way

And the main objectives:

- To work on personalisation, adaptive learning and run-time tracking (monitoring)
- To work on locpers-properties and loc-properties, data-type real, initialization
- To show and hide layers with different content depending on real time results
- To store and calcultate values depending on choices

- To set-up key activities to follow the learning proces depending on the set-up and on the feedback acquired
- To monitor the progress of the participants in real-time

The key parts in coding are:

Two roles are defined, one staff type (Teacher) and one learner type (Participant):

```
<roles>
   <learner identifier="student">
         <title>Participant</title>
   </learner>
   <staff identifier="teacher">
         <title>Teacher</title>
   </staff>
</roles>
```

Every property can be defined in the setttings.xml file:

```
Title of questionnaire<set-property
xmlns="http://www.imsglobal.org/xsd/IMS-LD_v1p0" ref="title" property-
of="self" view="value"/>

Points per right answer<set-property
xmlns="http://www.imsglobal.org/xsd/IMS-LD_v1p0" ref="points-right"
property-of="self" view="value"/>

Message of welcome<set-property xmlns="http://www.imsglobal.org/xsd/IMS-
LD_v1p0" ref="welcome" property-of="self" view="value"/>

Level 0 to, inclusive<set-property
xmlns="http://www.imsglobal.org/xsd/IMS-LD_v1p0" ref="Level0-to"
property-of="self" view="value"/>

Message feedback Level 0<set-property
xmlns="http://www.imsglobal.org/xsd/IMS-LD_v1p0" ref="msg-feedback-0"
property-of="self" view="value"/>

Level 1 to, inclusive<set-property
xmlns="http://www.imsglobal.org/xsd/IMS-LD_v1p0" ref="Level1-to"
property-of="self" view="value"/>
```

```
Message feedback Level 1<set-property
xmlns="http://www.imsglobal.org/xsd/IMS-LD_v1p0" ref="msg-feedback-1"
property-of="self" view="value"/>
…
```

For each question, a set of answers are defined using a group of properties called Q1_set, in the case of the first question. Also, the right answer is assigned to Q1_Right, and the related question is assigned to Q1. Every group of question and answers is configured in an independent XML file:

```
<set-property ref="Q1_Right" property-of="self" view="value"/>
<set-property-group ref="Q1_set" property-of="self" view="value"/>
<set-property ref="Q1" property-of="self" view="value"/>
```

The feedback and the questions are shown in a similar way to the previously depicted Unit of Learning GeoQuiz 3. There is also a XML file to monitor the current content of the properties, including the state for every participant supported-person:

```
Maximum of points<view-property ref="maximum" property-of="student"
view="value"/>
Total points earned<view-property ref="total" property-of="supported-
person" view="value"/>
Number of questions<view-property ref="number-questions" property-
of="student" view="value"/>
Average per question<view-property ref="average" property-of="supported-
person" view="value"/>
Accuracy<view-property ref="accuracy" property-of="supported-person"
view="value"/>
```

Before that the questionnaire go public the teacher/learning designer can monitor the current state of the questions and the answers, using the setting-ok.xml file. The teacher/learning designer must agree on the publication, switching on the property setting-ok:

```
<set-property ref="setting-ok" property-of="self"/>
```

This monitoring service has to be defined in the manifest file, inside a support activity:

```
<support-activity identifier="SA-setting-ok" isvisible="true">
   <title>Preview and publishing the questionnaire</title>
   <activity-description>
         <item identifier="I-SA-setting-ok" identifierref="res-setting-
ok" isvisible="true" />
   </activity-description>
```

```
    <complete-activity>
          <when-property-value-is-set>
                <property-ref ref="setting-ok" />
                <property-value>1</property-value>
          </when-property-value-is-set>
    </complete-activity>
</support-activity>
```

Once that the questionnaire is been published, the teacher-learning designer can monitor the students´ progress:

```
<environment identifier="env-monitoring">
    <title>Monitoring</title>
    <service identifier="service-monitoring">
          <monitor>
            <role-ref ref="student" />
            <title>Monitoring progress of students</title>
            <item identifierref="res-monitoring" />
          </monitor>
    </service>
</environment>
</environments>
```

The most important innovation in this manifest is the definition of the method, with every role-part providing a different action:

`Rolepart-1` provides the general set-up

```
<role-part identifier="rolepart-1">
    <title>Teacher</title>
    <role-ref ref="teacher" />
    <activity-structure-ref ref="setting-up" />
</role-part>
```

`Rolepart-2` provides the configuratiobn of questions and answers

```
<role-part identifier="rolepart-2">
    <title>Student</title>
    <role-ref ref="student" />
    <activity-structure-ref ref="questionsandanswers" />
</role-part>
```

`Rolepart-3`, `rolepart-4` and `rolepart-5` provides the different post-activities `flow1`, `flow2` and `flow3`

```
<role-part identifier="rolepart-3">
   <title>Student</title>
   <role-ref ref="student" />
   <learning-activity-ref ref="flow1" />
</role-part>
```

Rolepart-SA-settink-ok provides the teacher with the monitoring service of the configuration and the agreement for publishing the questionnaire

```
<role-part identifier="rolepart-SA-setting-ok">
   <title>Teacher</title>
   <role-ref ref="teacher" />
   <support-activity-ref ref="SA-setting-ok" />
</role-part>
```

Rolepart-monitoring provides the teacher with the service of monitoring the students´ performance

```
<role-part identifier="rolepart-monitoring">
   <title>Teacher</title>
   <role-ref ref="teacher" />
   <support-activity-ref ref="SA-monitoring" />
</role-part>
```

When the questionnaire is set-up it can be filled in by the studens in the same way as the follow GeoQuiz 3, also with the same mechanism for feedback and the next activity to carry out afterwards. The difference between both Units of Learning is on the online re-configuration of the questionnaire, even if the student has started, without republishing the whole UoL.

10.4.1. Remarks on this case study

Note about incremental remarks: from now onwards, the remarks from each case study will be built upon the remarks on the previous case studies, so that we don´t revisit the same issues unnecessarily:

- The monitoring service doesn´t cover any kind of user grouping. Therefore, a user cannot follow the performance of several people at the same time
- The teacher/learning designer cannot change the amount of questions or the answers. It is a fixed number for each entry
- Questions and answers are not personalised for user, and they are the same for everyone
- The communication between teacher and student is little. They can only see the values of properties but there is no other communication service between them

10.5 . **Case study, adaptation on the learner´s decision: JazzUK**

In this Unit of Learning a student can follow a course about Jazz and can choose two different itineraries, thematic and historic, based of his preferences (Figure 84). Also, some actions of monitoring can be tackled in the way described previously. This study was used as a base to explain the concept of adaptation with IMS-LD in Section 10.5. This a very complex Unit of Learning feeded with actual content for a real course, originally developed with EML. Several features on adaptation are shown, like monitoring services, hide/show classes, hide/show learning structures, quizzes, adaptive environments, et cetera.

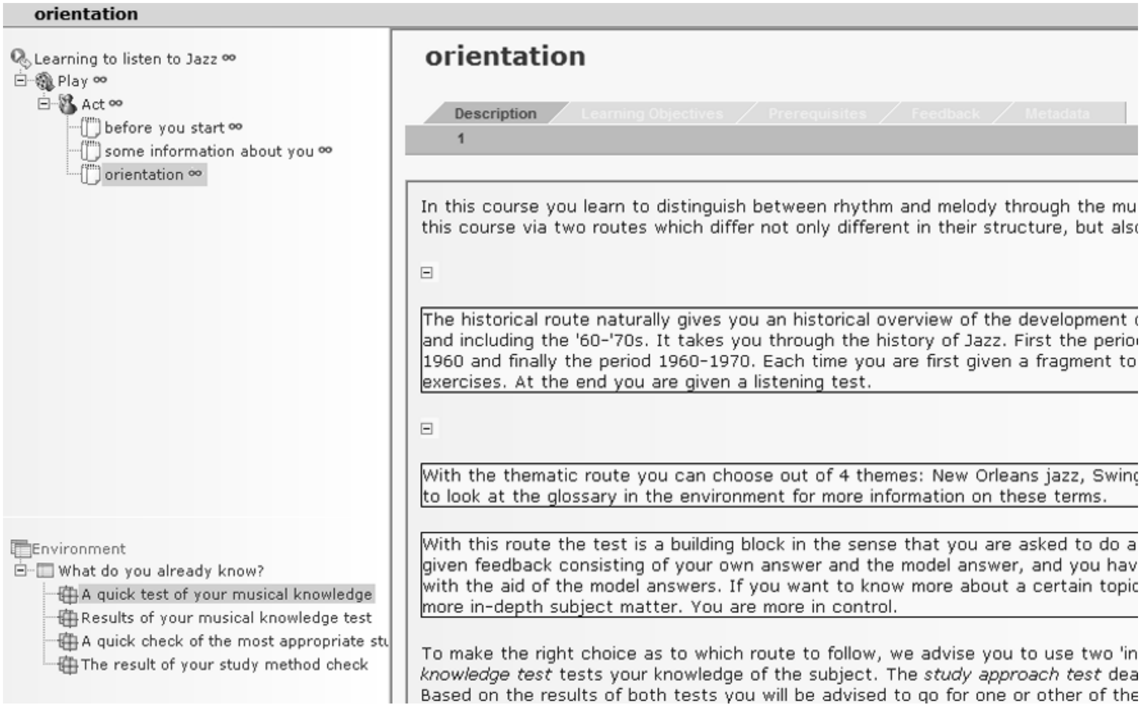


Figure 84. Learning to Listen to Jazz

Concerning the concept of adaptive learning a property called `'LP-choose-itinerary'` is set up to know whether the user has chosen one of the two itineraries. Each of them is described inside their Activity Structure, `'AS-historic'` and 'AS-thematic', previously defined in the manifest and out of the scope of this paper. All the process to choose an itinerary is programmed as a flow of conditions, taking one option or the another depending on the value of this property. Both Activity Structures are hidden in the beginning, when any value is inside the property yet:

```
<conditions>
<if>
  <no-value>
    <property-ref ref="LP-choose-itinerary"/>
  </no-value>
```

```
</if>
<then>
  <hide>
    <activity-structure-ref ref="AS-thematic"/>
    <activity-structure-ref ref="AS-historic"/>
  </hide>
</then>
```

The student can take one of the two options, ‘historic’ and ‘thematic’, available in a combo box. If the user takes the option ‘thematic’, his related structure ‘AS-thematic’ is shown and the non-related structure ‘AS-historic’ is hidden. The same things happen the other way around:

```
<if>
  <is>
    <property-ref ref="LP-choose-itinerary"/>
    <property-value>thematic</property-value>
  </is>
</if>
<then>
  <show>
    <activity-structure-ref ref="AS-thematic"/>
  </show>
  <hide>
    <activity-structure-ref ref="AS-historic"/>
  </hide>
</then>
```

These two different structures are able to have also non-identical contents or the same ones re-organized in several ways dealing with two complementary or opposite approaches and all is managed inside the same manifest coming with the Unit of Learning.

The input point to choose the type of itinerary is perfectly defined and it is always fixed and assigned to a different role-part:

```
<role-part identifier="id-1.5.3.8">
   <role-ref ref="student"/>
   <activity-structure-ref ref="As-s-thematisch"/>
</role-part>
<role-part identifier="id-1.5.3.10">
   <role-ref ref="student"/>
   <activity-structure-ref ref="As-s-historisch"/>
```

```
</role-part>
```

```
<activity-structure identifier="As-s-historisch" structure-
type="sequence">
   <title>historical route</title>
   <information>
         <item identifierref="res-1.5.2.1.1.1"/>
   </information>
   <environment-ref ref="env-1.5.2.1.1.1"/>
   <learning-structure-ref ref="la-bebop"/>
   <learning-activity-ref ref="la-free-jazz"/>
   <learning-activity-ref ref="la-change-itinerary-question"/>
   <learning-activity-ref ref="la-swing"/>
   <learning-activity-ref ref="la-new-orleans"/>
</activity-structure>
```

```
<activity-structure identifier="As-s-thematisch" structure-
type="selection" number-to-select="5">
   <title>Thematic route</title>
   <information>
         <item identifierref="res-1.5.1.1.1.1"/>
   </information>
   <environment-ref ref="env-1.5.1.1.1.1"/>
   <learning-structure-ref ref="la-bebop"/>
   <learning-activity-ref ref="la-free-jazz"/>
   <learning-activity-ref ref="la-change-itinerary-question"/>
   <learning-activity-ref ref="la-swing"/>
   <learning-activity-ref ref="la-new-orleans"/>
</activity-structure>
```

In this last activity structure type **selection** the user needs to complete 2 activities (user-choice) before he/she gets the question that allows for changing the itinerary, as there is no sequence where to launch the question from, but an specific amount of activities carried out (2 activities). Therefore, we need to define a property for each learning activity in the activity structure that sums 1 when it is completed and remains 0 when it is not. For instance, for the learning activity free-jazz:

```
<locpers-property identifier="prop-free-jazz">
   <title>free jazz thematically complete</title>
   <datatype datatype="integer"/>
   <initial-value>0</initial-value>
```

```
</locpers-property>
```

```
<learning-activity identifier="la-free-jazz" isvisible="true">
   <title>free jazz</title>
   <activity-description>
         <item identifierref="res-free-jazz">
               <title>what is Free Jazz</title>
         </item>
   </activity-description>
   <complete-activity>
         <user-choice/>
   </complete-activity>
   <on-completion>
         <change-property-value>
               <property-ref ref="prop-free-jazz"/>
               <property-value>1</property-value>
         </change-property-value>
   </on-completion>
</learning-activity>
```

Later on, we define a condition to permanently check the state of these properties, to sum up their values, and to show the expected question as long as the final result is greater than 1, so it is actually 2 or greater, what represents the condition to ask the question itself. As the learning activities are the same in both itineraries we have also to check the value of the property prop-itinerary to know whether the user is following the right itinerary to run this condition (thematic):

```
<conditions>
   <if>
         <and>
         <is>
               <property-ref ref="prop-itinerary"/>
               <property-value>thematic</property-value>
         </is>
         <greater-than>
               <sum>
                     <property-ref ref="prop-bebop"/>
                     <sum>
                           <property-ref ref="prop-new-orleans"/>
                           <sum>
                                 <property-ref ref="prop-swing"/>
```

```
                                  <property-ref ref="prop-free-jazz"/>
                            </sum>
                      </sum>
                </sum>
                <property-value>1</property-value>
          </greater-than>
          </and>
    </if>
    <then>
          <show>
              <learning-activity-ref ref="la-change-itinerary-question"/>
          </show>
    </then>
    <else>
          <hide>
              <learning-activity-ref ref="la-change-itinerary-question"/>
          </hide>
    </else>
</conditions>
```

10.5.1. **Remarks on this case study:**

- There is a lack of flexibility on the input point to raise the question and possibly change the itineraries. In the type sequence the learning activity with the question is always at the same place. In the type selection, the question is always asked after 2 completed learning activities. What if the learning designer/teacher wants to change the method and shift this input point? From 2 to 3 completed LA´s or from the third in a row to the forth?
- There is no chance to run the Unit of Learning (the whole UoL or a part, such a Learning Activity) two times within the same instance. Once that a Learning Activity is closed, the user can read it again but the associated learning flow cannot be executed. For instance, after that the question to change the itinerary is made in the historisch-route, there is no way to come back
- This UoL is a real course. There is no flexibility to change the content either. When the teacher/learning designer wants to keep the same method and the same structure, but he/she only wants to change one single HTML page with some content, the UoL has to be validated and published again, the learner and the teacher have to be enrolled and the learning process starts from the very beginning

- There is no chance to handle absolute time to start the course and-or a specific activity with. Only relative time to the precise time when the instance is created out of the UoL

10.6 . Case study, adaptation on the teacher´s decision: FSA

In this case study the teacher monitors the whole process of assessment, since one essay is sent until the final grade is published, through a discussion of two steps with the student who sent the task (Figure 85).

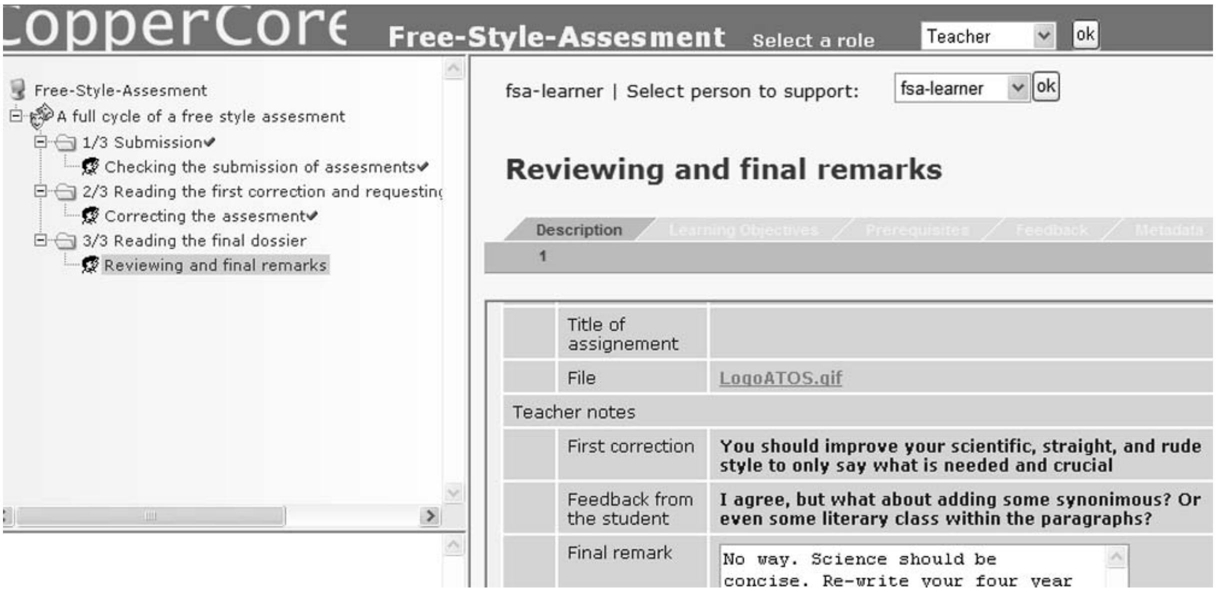


Figure 85. Free Style Assessment

The learning flow is as follows:

- The student submit his personal info and his assesment to the tutor
- The teacher check the submission of all the students and close this activity
- The teacher grades the assesment and make his own remarks
- The student read the first remarks and can provide some feedback (critic, complain, remark...). The teacher closes this activity
- The teacher grades (second round) the assesment, taking into account student´s remarks, and assign a qualificationa and a numerical mark. The teacher closes this activity
- The student read his dossier and can also have a look to the marks and qualifications of others

10.6.1. **Remarks on this case study:**

- There is no chance to make a connection to an already existing DataBase (for instance, to make a query or to import already enrolled students or teachers). The data type of connection is not supported. Therefore, every enrolment has to be done by hand or running a specific tool for that
- Furthermore, any connection with the external world is impossible (as we addressed in the previous chapter on Interoperability). For instance, a real-time effective communication between a LMS and an IMS-LD UoL is no possible, so far, so that they cannot benefit each other from mutual services and resources. There is no dispatcher allowing such this connection
- A file uploaded from the hard disk of the computer of a user is stored in a file-type property inside the internal database of the engine (CopperCore in this case). There is no chance to change the by default configuration on storing or retrieving resources. There is not any facility to manage such these uploads either. Although this is an issue concerning tools, the core documents of IMS-LD don´t provide with this information and-or facility either
- There is a fixed amount of iterations in the reviewing process. The teacher/learning designer can make this amount smaller but he/she cannot make it bigger ever. What happens if the discussion needs one additional round?
- There is no chance no make a dynamic selection of users in order to create groups. The teacher can monitor what happens one by one, and he/she can also provide some feedback to one by one. We could set-up a property to answer by groups, but these groups should be established before the actual start. However, if the teacher wants to make a dynamic creation of a group of students depending on their answers, this is not possible so far

10.7 . **Case study, adaptation on the learner´s behaviour: PET**

We show an example of gathering information based on the learner´s behaviour: the Planning Educational Task (PET). The intention on describing the very basics of this experiment is to understand its general drives and structure, in order to make an analysis concerning IMS-LD. This experiment is based on feedback and adaptation [321], one of the topics already approached in previous sections. Planning Educational Task [322], which simulated the planning of speakers for a conference, existed in two versions, a feedback version, and a no feedback version. The feedback version provided visual destination feedback related to the learner’s actions and moves sofar. In the Planning Educational Task, students can first be expected to start to explore the application and in the meanwhile work towards the imposed goal: solving the problem. A routine or strategy will not be available in the beginning. Therefore, students will need to explore and discover in a probably non-structured manner, which can be compared with Prensky's [127] statement that in games, one of the most usual techniques is trial-and-error, defined as the absence of a systematic strategy when a learner plays [128]. The Planning Educational Task focused on the opposition between externalization and internalization of information in the interface, corresponding to the difference between respectively providing and hiding visual feedback as long as the player tries to solve the problem. This feedback is fostering orientation on what to do next, it is guiding the player in the sense that it shows which choices are available. However, when moves are made, the player is at all times allowed to undo the taken action(s) and to go backwards to establish a new strategy to follow. This strategy is partially based on trial-and-error movements, although the level of risk that a player takes in every movement could be different depending on the level of provided feedback, as we show in the coming sections.

The software (Macromedia Flash MX [323]) logged all the moves that participants made. The Conference Planner was developed by The Open University of The Netherlands and funded by the European UNFOLD Project [324]. It consists of four different components. The first one is the dynamic interface that shows each set of demands for a conference and allows the end-user to solve the problem in an easy way, based on drag & drop movements. The second one is the core of the application itself: the set of rules and related algorithms. Here is defined which actions are permitted, based on the requirements of the experiment and which are the subsequent consequences. The third component is a database, with all the scenarios used in the experiment. The forth component is the logging-module that writes all the clicks and drag & drop moves and their associated times, as well as waiting times to an external spreadsheet. The logs provide data for analysis of the results.

In the experiment, the students had to solve 5 different conference scheduling situations. The conference speakers each had different demands, and they had to be scheduled into one of three available rooms (each with its own facilities and availability).

The difference between feedback and no feedback was implemented by highlighting all legal slots in the feedback version where a person can be placed. In this version (Figure 86), when one clicks on a speaker in the list on the left, the legal slots (those satisfying the constraints and being available) in the timetable turned green. Note that this does not show the best slot to place a speaker, but simply which slots are possible. To move a speaker from the left to a slot on the right, the little boxed icon in front of each speaker's name had to be picked up and dragged to its destination slot with the mouse. Not all the timeslots in the grid are always available. Some where unavailable all the time, indicated with light-gray, for example the timeslots during lunchtime (13:00), but also some arbitrary other slots. The empty available timeslots were shown in white, and the ones that were already occupied by a speaker would display the name of a speaker.

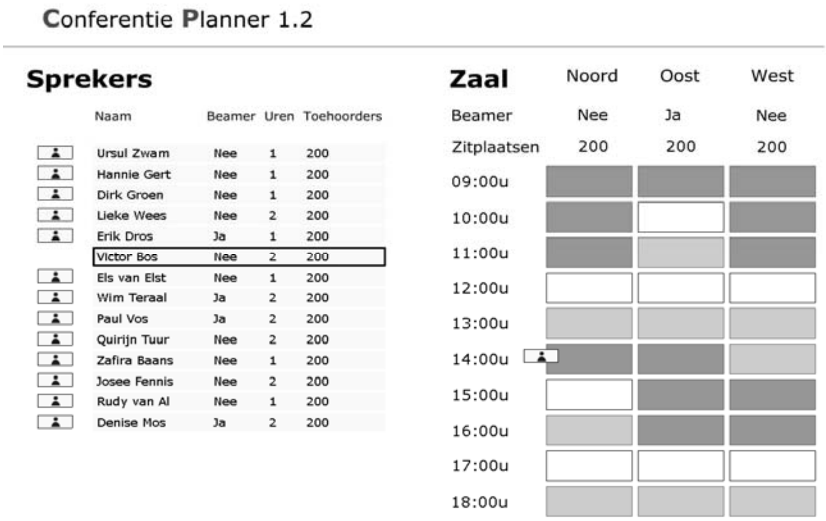


Figure 86. Conference Planner in the feedback version when a speaker was picked up, the legal timeslots turned green

In the internalization condition the green feedback was absent, and one has to look up information and constraints by one self all the time (Figure 87). No other differences existed between the two conditions.

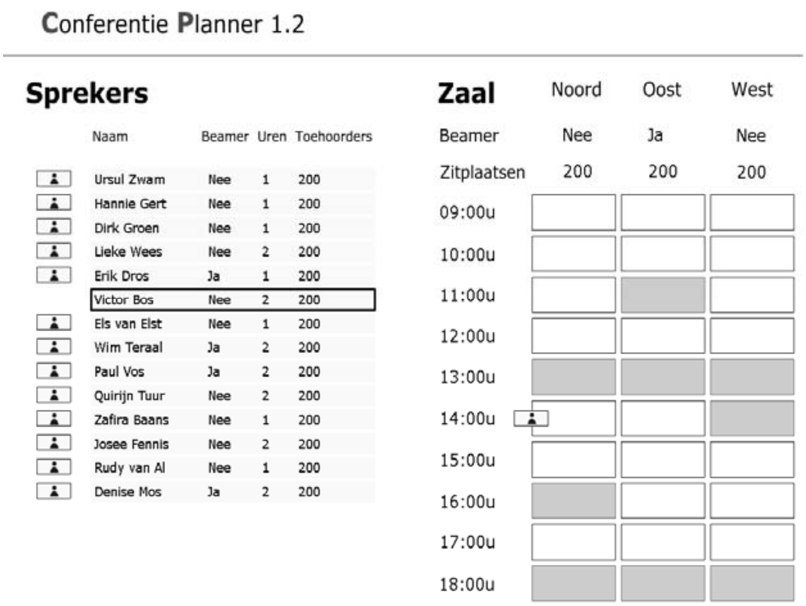


Figure 87. Conference Planner in the no feedback version

In both conditions, a list of speakers that had to be scheduled was given on the left. Each speaker had his/her own constraints displayed next to them, which could vary on a maximum of three variables:

- Beamer* needed (yes/no, in Dutch: ja/nee) *(*Dutch/German pseudo-anglicism meaning 'projector'*)
- Number of hours they will speak (1 or 2, in Dutch: uren)
- Number of expected listeners (in Dutch: toehoorders)

The assignment was to place all the speakers on the schedule timetable, while taking the different constraints into account. A solution where each speaker was scheduled (and all the constraints were met) always existed. The students had to perform 5 tasks in which the rooms, the speakers and their constraints varied. During the study, we collected the following time-based and move-based measures:

- Total time needed: the average time needed to solve the tasks.
- Time before first move: the time between the moment the problem appears on-screen and the first move. It is an indicator for planning, telling how long subjects analyzed the problem before they started solving it.
- Inter-move latency: the time that passes between having placed a speaker, and picking up the next. We interpret this measure as a planning indicator.
- Superfluous moves: the problems have a shortest path solution, with an optimal amount of moves (speakers dragged from left to right) to solve them. E.g. in the case of a list with 15 speakers, the optimal solution would be 15 moves (placing all speakers correctly at once). Superfluous moves are all the extra unnecessary actions besides this shortest path. We use this measure as the main performance measure, because it reflects the efficiency with which the task has been solved.
- Knowledge: assessment afterwards with essay questions about pictures of situations that were either legal or illegal given the constraints. It had to be decided whether or not certain situations could occur, and why (or why not).
- Strategy: we looked (per task) at whether or not subjects started solving the problem with the best strategy, by first moving the speakers that had the most stringent constraints. When this is done, one can assume that some amount of planning has happened.

The aim of recording all this information is to provide some further adaptive guidance on the learning process of the user. Some results came up with the experiment. However, they are out of the scope of this dissertation. All of them can be found at [321; 325]

10.7.1. **Remarks on this case study:**

We planned to make this experiment with IMS-LD. We found a number of obstacles that didn´t allow us for developing it:

- The connection with an external DataBase, aside from the internal DataBase of the engine (CopperCore 3.0) is not possible so far and it is not in the scope of the specification
- IMS-LD and the DataBase of the engine don´t allow for any kind of query from recorded information
- IMS-LD doesn´t allow for recording the behaviour of the user, so that none of measures listed above (i.e., Total Time Needed, Time Before First Move) can be retored or retrieved
- Users cannot be dynamically enrolled within the Unit of Learning, once it has started, and they are managed by an external tool
- IMS-LD doesn´t allow for saving information into external files or for retrieving information from any external source
- We conclude that we were looking for an example of adaptation based on the user´s behaviour and we found that this cannot be developed with IMS-LD. We also realized that, even when the executable module is developed with other technologies (Macromedia Flash and PHP, for instance) it cannot be integrated with IMS-LD in any way. Therefore, we also found an interoperability problem. Although IMS-LD is not developed thinking of such this kind of interactivity with users, it could allow for a valid integration with external resources using a layer of communication/dispatcher

11. Annex III: Glossary and related links

11.1. Glossary

(First published at [53])

Act

Part of a play. Roles are played by those taking part, for example learner, tutor, mentor, and so on.

http://www.cetis.ac.uk/lib/media/WhatIsLD_web.pdf

Activity structure

A container for activities and/or other activity-structures allowing sequencing and selection of its elements, and assigned to a role at a particular point in the learning process. http://www.imsglobal.org/af/afv1p0/1523626

Collaborative Learning

Performing learning activities with two or more participants, including both asynchronous (e.g., discussion board) and synchronous (e.g., chat, video conferencing) systems. Standards based descriptions of collaboration require descriptions of the participants involved (and their different roles), the content involved, and the systems required to facilitate collaboration (cf. Learning Design WG). http://www.imsglobal.org/af/afv1p0/1523626

Collaboration

A joint effort, facilitated by network technology with email, FTP (i.e., File Transfer Protocol used for downloading files on the Internet), and more advanced means of sharing ideas, documents, and data. http://www.trinity.edu/~rjensen/245glosf.htm

Community

A dynamic whole that emerges when a group of people share common practices, are interdependent, make decisions jointly, identify themselves with something larger than the sum of their individual relationships, and make a
long term commitment to the well-being of their own, of others and of the group as a whole (Shaffer and Anundsen, 1993)

Community of practice

Group of practitioners in a certain field of expertise who share knowledge and experience. The group members typically possess different levels of experience and novices are supported to become expert members of the group through some guidance (or scaffolding) mechanisms (Lave and Wenger, 1991).

Components

The collection of parts that is reusable within a learning design. The elements role, activity-structure, learning-activity, support-activity and environment are all included in the components section of an IMS Learning Design document instance. http://www.imsglobal.org/af/afv1p0/1523626

Condition

A rule used to influence for flow of a play in a Unit of Learning. Used in conjunction with properties, conditions add further refinement and personalisation facilities to a learning design. Conditions have the basic format:

IF [expression]

THEN [show, hide, or change something or notify someone]

The expressions are mostly defined on properties (e.g., IF pre-knowledge-English="4"). http://www.imsglobal.org/af/afv1p0/1523626

Design patterns

Design patterns provide a structure for integrating the analysis and solution of a problem, in a way that is sensitive to context and informed by theory and evidence. A pattern suggests, rather than prescribes, a solution (http://dspace.learningnetworks.org/bitstream/1820/300/2/e-len+design+patterns+booklet_final.pdf)

Dossier / Portfolio

Lifelong learners have specific expertise and competence in a discipline and these must be registered and updated in a
learning dossier. The competence and expertise levels stored in the dossier must be standardized to be able to position a learner in an LN. http://dspace.learningnetworks.org/bitstream/1820/32/4/from-journal-LR_71-92.pdf

Educational Modeling Language

Semantic information model and binding, describing the content and process within a "unit of study" from a pedagogical perspective in order to support re-use and interoperability (Rawlings et al., 2002).

ELearning

Learning, in which computers and the Internet play an important role in the delivery, support, administration and assessment of learning (Kirschner and Paas, 2001).

Facilitators

In a Learning Network facilitators manage the operation of the network, they are for instance moderators and webmasters
(Ferber, 1999; http://dspace.learningnetworks.org/bitstream/1820/66/2/framework-TDprogramme-internalUse.pdf).

Feedback-First order-Second order

-First-order feedback means that people in the community know what their counterparts are doing or have done regarding the UoLs in the network. This provides information for navigation and behavioral models within the community.
-Second order feedbackrefers to feedback about the emergent properties in the system: what is the performance of the community and how it is organized (Gilbert, 1995)

Global elements

A mechanism used in order to be able to set and view properties during the teaching and learning. There are four global elements: set-property, view-property, set-property-group and view-property-group. Global elements are designed to be included in any XML content schema by use of XML namespaces (e.g., for inclusion in XHTML). http://www.imsglobal.org/af/afv1p0/1523626

IMS Learning Design

Specification used to describe learning scenarios. It allows these scenarios to be presented to learners online, and enables them to be shared between systems. It can describe a wide variety of pedagogical models, or approaches to learning, including group work and collaborative learning. It does not define individual pedagogical models; instead it provides a high level language, or meta-model, that can describe many different models. The language describes how people perform activities using resources (including materials and services), and how these three things are coordinated into a learning flow (http://www.cetis.ac.uk/lib/media/WhatIsLD_web.pdf). IMS-LD enables the

representation of the learning and teaching processes in a UoL to be interoperable and machine interpretable. It provides a framework for including learning activities, support activities, assessment and learning or knowledge resources. IMS-LD can express the pedagogical approach taken in the UoL, and supports personalisation of learning routes and reusability (Koper & Van Es, 2004).
http://dspace.learningnetworks.org/bitstream/1820/32/4/from-journal-LR_71-92.pdf

Learning activity

An activity to be carried out by a learner in order to obtain a learning objective. The notion of a Learning Activity recognizes that learning can happen with or without learning objects (learning is different from content consumption) and that learning comes from learners being active. http://www.imsglobal.org/af/afv1p0/1523626

Learning community

(a) has distributed control,

(b) shows commitment to the generation and sharing of new knowledge, (c) learning activities are flexible and negotiated,

(d) community members are autonomous,

(e) shows a high level of dialogue, interaction and collaboration, and (f) there is a shared goal, problem or project that brings common focus and incentives to work together. http://carbon.cudenver.edu/~bwilson/dlc.html

Learning design method

Determines which roles get which type of activity at a given time, based on a pedagogical approach.
http://dspace.learningnetworks.org/bitstream/1820/32/4/from-journal-LR_71-92.pdf

Learning environment

A social system focused on the permanent development and certification of human knowledge and competencies in a particular domain. http://dspace.learningnetworks.org/bitstream/1820/38/2/koper-inaugural-address-eng.pdf

Learning object

Any entity, digital or non-digital, that can be used, reused or referenced during technology-supported learning

Learning objectives

The intended outcome for learners. It is possible to define learning objectives both at the global level of the Unit of Learning and for every single learning activity in the learning design. http://www.imsglobal.org/af/afv1p0/1523626

Learning scenario

Learning design which contains play, act, and role-parts elements (analogous to a theatrical play). http://www.cetis.ac.uk/lib/media/WhatIsLD_web.pdf

Learning Technologies

Means of formalizing pedagogical and organizational thinking in such a way that it can be implemented in a technical domain. That is, they bridge the gap in *eLearning* between educational, organizational, and technical requirements (Jochems, van Merriënboer, and Koper, 2004).

Levels of implementation within learning design

Level A, with the definition of the method, plays, acts, roles, role-parts, learning activities, support activities and environments. It is the core of the specification, contains the description of the elements that configure IMS-LD and the coordination between them. For instance, role-parts define what activities must be taken by a role in order to complete an act and, subsequently, a play.

Level B adds properties, conditions, monitoring services and global elements to Level A, and provides specific means to create more complex structures and learning experiences. Properties can be used as variables, local or global ones, storing and retrieving information for a single user, a group or even for all the characters involved. Through these mechanisms the learning flow can changed at the run time, as decisions can be made taking into account dynamic content.

Level C offers the opportunity for more sophisticated learning designs through notifications (messaging), which allow for notification of new activities to be triggered automatically in response to events in the learning process. It enables the automation of learning flow activities, which are triggered by the completion of tasks, rather than the learning flows being pre-planned. For instance, a teacher may be notified by email that an assignment has been submitted and needs marking; once the score has been posted, the learner may be notified to undertake a new activity according to the result.
http://www.cetis.ac.uk/lib/media/WhatIsLD_web.pdf

Monitoring service

The monitor service provides a facility for users to look at their own properties or that of others in a structured way. A monitor uses global properties in resources of type 'imsldcontent' to view the properties of one-self or of all users in a role. http://www.imsglobal.org/

Notification

The triggering of new activity or the sending of a message in response to an event. Events which trigger notifications include the completion of an activity and the changing of a property-value. http://www.imsglobal.org/af/afv1p0/1523626

Pedagogical model

-Prescribes an effective teaching/learning process for a class of learners to achieve a class of learning objectives in a class of situations

-The most effective learning products

or environments are those that are problem-centred and involve the student in four distinct phases of learning:

(1) activation of prior experience,

(2) demonstration of skill,

(3) application of skill and

(4) integration of these skills into real-world activities'. He further summarizes the underlying 'first principles of instruction' by stating that learning is promoted when: learners are engaged in solving real world problems; existing knowledge is activated as the foundation for new knowledge; new knowledge is demonstrated to the learner; new knowledge is applied by the learner; and new knowledge is integrated into the learner's world.

(Merrill, 2003)

Play

A play specifies which roles perform what activities in what order. A play is modelled according to a theatrical play with acts and role-parts. In general: a play consists of a sequence of acts. In each act, different activities are set for different roles and are preformed in parallel. When an act is completed, the next act starts until the completion requirements for the learning design are met.

http://www.imsglobal.org/af/afv1p0/1523626

Property

A variable used for a variety of purposes including monitoring, personalisation and assessment. Learning Design supports five types of properties: local properties, local-personal properties, local-role properties, global-personal properties and global properties.

http://www.imsglobal.org/af/afv1p0/1523626

Providers

Can be educational institutions, companies and libraries that provide lifelong learners (e.g. employees), the learning services (e.g. tutoring services) or the learning resources (e.g. books, CDs). http://dspace.learningnetworks.org/bitstream/1820/32/4/from-journal-LR_71-92.pdf

Support activity

An activity carried out in support of a role performing one or more learning activities. For example, a staff role might have the support activity to grade reports made by people in the learner role named 'student'. Each student creates his/her own report and the tutor grades every report (repeating the 'grade report' support activity). http://www.imsglobal.org/af/afv1p0/1523626

Unit of Learning (UoL)

Complete piece of educative work created following a learning design structure and packaging the related resources, web links and several learning material and services in only one ZIP file. Therefore, it is a compressed file with a) a XML manifest, describing method, plays, acts, roles, activities, environments, properties, conditions and/or notifications of the Learning Design specification and also pointing to the related resources; and b) a set of files or resources mentioned in the XML manifest."

Run of Unit of Learning

Instantiation for a specific set of learners in a certain time frame (e.g., a class, the actual run of a workshop).

http://dspace.learningnetworks.org/bitstream/1820/32/4/from-journal-LR_71-92.pdf

Usability

An LN is usable when it supports rapid learning, high skill retention, low error rates and high productivity. It is consistent, controllable and predictable, making it pleasant and effective to use (Preece, 2000).

Use cases

Abstractions of scenarios in which the concrete behaviour of persons within a system, or using a system is described

11.2 . Links and resources

IMS–IMS-LD

- IMS Consortium
http://www.imsglobal.org/
- IMS Learning Design Specification
http://www.imsglobal.org/learningdesign/index.html
- **UNFOLD** Project
http://www.unfold-project.net
- Learning Networks
http://www.learningnetworks.org
- Learning Network for Learning Design
http://moodle.learningnetworks.org/
- Runnable Example Units of Learning
http://moodle.learningnetworks.org/course/view.php?id=20
- Learning Networks Dspace
http://dspace.learningnetworks.org/index.jsp
- The Valkenburg Group
http://www.valkenburggroup.org/valkenburggroup-org.htm
- R2R: Learning Design
http://commons.ucalgary.ca/weblogs/learningdesign/

Tools

Editors

- Alfanet IMS-LD
http://dspace.learningnetworks.org/handle/1820/103
- Copperauthor
http://www.copperauthor.org/
- eLive
http://www.elive-IMS-LD.com/content/index_ger.html
- Eduplone Learning Sequencer
http://eduplone.net/index_html?cl=en
- MOTPlus
http://www.licef.teluq.uquebec.ca/gp/eng/productions/mot.htm

- RELOAD
http://www.reload.ac.uk/
- SchulCMS
http://www.schulcms.de

Players

- CopperCore Learning Design Engine
http://www.coppercore.org/
- RELOAD
http://www.reload.ac.uk/
- SLED
http://sled.open.ac.uk/web/

Virtual Learning Environments

- LAMS Learning Activity Management System (Level A export)
http://www.lamsinternational.com/
- .LRN (full IMS-LD compliance under development)
http://www.dotlrn.org/
- Moodle
(IMS-LD export-import under development)
http://moodle.org/
- NetUniversité (under development)
http://www.cepiah-hds.utc.fr:8080/CEPIAH/web/index.jsp

Others

- ACETS
Investigation of the pedagogical use of reusable learning objects
http://www.acets.ac.uk/
- ALFANET
Set of components for learning providers using personalisation and adaptation
http://rtd.softwareag.es/alfanet/
- DialogPlus
Helps teachers to define learning activities through a taxonomy with IMS-LD export
http://www.dialogplus.org/
- ELF (*ELearning* Framework)
Common approach to Service Oriented Architectures for education

http://www.elframework.org/

- LearningMapR

Pedagogical design tool using IMS-LD templates

http://lt3.uwaterloo.ca/innovation/ldrg.html

Institutions

- Centre for Educational Technology Interoperability Standards (CETIS)

http://www.cetis.ac.uk

- Information Society Technologies

http://www.cordis.lu/ist/

Other IMS Specifications

- IMS Content Packaging

http://www.imsglobal.org/content/packaging/index.html

- IMS Learner Information Packaging

http://www.imsglobal.org/profiles/index.html

- IMS Question and Test Interoperability

http://www.imsglobal.org/question/index.html

- IMS Simple Sequencing

http://www.imsglobal.org/simplesequencing/index.html

Related specifications

- Dublin Core Metadata Initiative

http://dublincore.org

- Advanced Distributed Learning. SCORM

http://www.adlnet.org/index.cfm?fuseaction=scormabt

- About SCORM

http://www.rhassociates.com/scorm.htm

References / Referencias

[1] IMSLD. (2003). IMS Learning Design. Version 1. Retrieved February 27th, 2004, from http://www.imsglobal.org/learningdesign/index.cfm

[2] Burgos, D., Tattersall, C., & Koper, R. (2006). Representing adaptive eLearning strategies in IMS Learning Design. *Proceedings of International Workshop in Learning Networks for Lifelong Competence Development, TENCompetence Conference*. March 30th-31st, Sofia, Bulgaria. Retrieved June 30th, 2006, from http://dspace.learningnetworks.org

[3] Vogten, H., & Martens, H. (2005). CopperCore 3.0. Retrieved April 13th, 2006, from www.coppercore.org

[4] Van der Vegt, W. (2005). CopperAuthor. Retrieved April 13th, 2006, from www.copperauthor.org

[5] Bolton. (2004). Reload Project. Retrieved April 16th, 2006, from www.reload.ac.uk

[6] OUNL. (2005). Learning Network for Learning Design. Retrieved March 6th, 2007, from http://imsld.learningnetworks.org

[7] OUNL. (2002). DSpace. Retrieved June 22nd, 2007, from http://dspace.learningnetworks.org

[8] Paquette, G., Marino, O., Teja, I. D. l., Lundgren-Cayrol, K., Lonard, M., & Contamines, J. (2005). Implementation and Deployment of the IMS Learning Design Specification. *Canadian Journal of Learning and Technology, 31*(2).

[9] Pacurar, E. (2005). netUniversité.

[10] Dougiamas, M. (2003). Moodle. Retrieved May 10th, 2006, 2006, from http://moodle.org

[11] ADL. (2004). Sharable Content Object Reference Model (SCORM) 2004. 2004. Retrieved December 8th, 2004, from http://www.adlnet.org/index.cfm?fuseaction=DownFile&libid=648&bc=false

[12] IMSSS. (2003). IMS Simple Sequencing. Retrieved May 10th, 2006, 2006, from www.imsglobal.org

[13] Dalziel, J. (2003). LAMS. Retrieved June 14th, 2007, 2007, from www.lamsfoundation.org

[14] DotLRN-Project. (2006). dotLRN. Retrieved June 22nd, 2007, from http://www.dotlrn.org/

[15] Burgos, D. (2006). *The structure and behavior of virtual communities engaged in informal learning about e-learning standards (Estudio de la estructura y del comportamiento de las comunidades virtuales de aprendizaje no formal sobre estandarización del e-learning).* Unpublished Doctoral dissertation, European University of Madrid, Villaviciosa de Odón, Madrid, Spain.

[16] Burgos, D., Berbegal, N., & Griffiths, D. (2005). IMS Learning Design Level 0. Retrieved August 7th, 2006, from http://imsld.learningnetworks.org/mod/resource/view.php?id=174

[17] Cetis. (2005). Centre for educational technology interoperability standards. Retrieved March 4th, 2007, 2007, from http://www.cetis.ac.uk

[18] Burgos, D., & Griffiths, D. (2006). E-learning specifications. An introduction. *Proceedings of eLearning Europe*: European Commission of Education and Culture

[19] IMSCP. (2001). IMS Content Packaging Specification. Version 1.1.3. Retrieved February 27th, 2007, from http://www.imsglobal.org/learningdesign/index.cfm

[20] IMSLIP. (2001). IMS Learner Information Package. Version 1. Retrieved December 8th, 2004, from http://www.imsglobal.org/profiles/index.html

[21] IMSQTI. (2000). IMS Question and Test Interoperability. Retrieved May 10th, 2006, 2006, from www.imsglobal.org

[22] IMSGlobal. (1997). IMS Global Consortium. Retrieved February 13th, 2008, from www.imsglobal.org

[23] Committee, I. L. T. S. (2002). IEEE LTSC Learning Object Metadata. Retrieved Jan 12th, 2008, from http://www.ieeeltsc.org/standards/1484-12-1-2002/

[24] DCMI. (1995). Dublin Core Metadata Initiative. Retrieved Jan 12th, 2008, from http://dublincore.org/

[25] Botturi, L., Derntl, M., Boot, E., & Figl, K. (2006). A Classification Framework for Educational Modeling Languages in Instructional Design. In Kinshuk, R. Koper, P. Kommers, P. Kirschner, D. G. Sampson & E. , Kerkrade, Netherlands. (Eds.), *Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT2006)*. Los Alamitos, CA: IEEE.

[26] Verdejo, M. F., Barros, B., Read, T., Mayorga, J. I., Vélez, J., & Calero, M. Y. (2003). Active document description. Technical Report n. 2. Retrieved May 20th, 2007, from http://sensei.lsi.uned.es/ActiveDocument/

[27] Verbert, K., & Duval, E. (2004). Towards a global architecture fore learning objects: a comparative análisis of learning object content models. In *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 202-208): Ed-Media.

[28] Laforcade. (2004). *Méta-modélisation UML pour la conception et la mise en oeuvre de situations-problèmes coopératives*: L'université de Pay e des Pays de l'adour

[29] Koper, R., & Manderveld, J. (2003). Educational Modelling Language: Modelling reusable, interoperable, rich and personalised units of learning. *The British Journal of Educational Technology*.

[30] Botturi, L. (2006). E2ML. A Visual Language for the Design of Instruction. *Educational Technologies Research & Development, 54*(3), 265-293.

[31] Martel, C., Vignollet, L., Ferraris, C., David, J.-P., & Lejeune, A. (2006). LDL: an alternative EML. In *Proceedings of ICALT 2006* (pp. 1107-1108). Kerkrade, The Netherlands.

[32] Slavin, R. (1995). *Cooperative learning: theory, research and practice*. Boston: Allyn & Bacon.

[33] De la Teja, I., Lundgren-Cayrol, K., & Paquette, G. (2005). Transposing MISA Learning Scenarios into IMS Units of Learning. *Journal of Interactive Media in Education (Advances in Learning Design Special Issue), 2005/14*.

[34] Rodríguez-Artacho, M., & Verdejo Maíllo, M. F. (2004). Modeling Educational Content: The Cognitive Approach of the PALO Language. *Educational Technology & Society, 7*(3), 124-137.

[35] Koch, M. (2002). Interoperable Community Platforms and Identity Management in the University Domain. *International Journal on Media Management, 4*(1), 21-30.

[36] Brickley, D. (1996). Towards and open question-interchange framework. Retrieved February 13th, 2008, from http://www.ilrt.bris.ac.uk/netquest

[37] Buendía, F. (2003). *Xedu: un modelo para la especificación de componentes estática y dinámica de una aplicación instructiva*: University of Valencia

[38] Griffiths, D., Blat, J., Garcia, R., Vogten, H., & Kwong, K. (2005). Learning Design Tools. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 109-136). Berlin-Heidelberg: Springer Verlag.

[39] Magee, M. (2005). E-Learning Lifecycles: How Communities and Context can affect E-learning Specifications and Tool Design. *Canadian Journal of Learning and Technology, 30*(3).

[40] McAndrew, P., Nadolski, R., & Little, A. (2005). Developing an approach for Learning Design Players. *Journal of Interactive Media in Education (Advances in Learning Design Special Issue), 2005/14*.

[41] Tattersall, C., Vogten, H., & Hermans, H. (2005). The Edubox Learning Design Player. In E. J. R. Koper & C. Tattersall (Eds.), *Learning Design...* (Vol. 19, pp. 303-310). Berlin: Springer-Verlag.

[42] OUUK. (2005). Sled player. Retrieved April 13th, 2006, from http://sled.open.ac.uk

[43] Alfanet. (2005). Alfanet Project. Retrieved December 28th, 2006, from http://alfanet.ia.uned.es/

[44] TENCompetence. (2005). TENCompetence Project. Retrieved July 31st, 2006, from www.tencompetence.org

[45] Miao, Y. (2005). CoSMoS: Facilitating Learning Designers to Author Units of Learning Using IMS LD. In C. K. Looi, D. H. Jonassen & I. Mitsuru (Eds.), *Proceediongs of the International Conference on Computers in Education* (pp. 275-282). Singapore: IOS Press.

[46] Karampiperis, P., & Sampson, D. (2004). Adaptive learning object selection in intelligent learning systems. *Journal of Interactive Learning Research, 15*(4), 389-408.

[47] Burgos, D., & Koper, R. (2005). Virtual communities, research groups and projects on IMS Learning Design. State of the art, key factors and forthcoming challenges. *E-Journal of Educational Research, Assessment and Evaluation, 11*(2).

[48] Burgos, D., & van Nimwegen, C. (2005). Planning Educational Task. from http://moodle.learningnetworks.org/course/view.php?id=45

[49] Pro-Learn. (2006). Pro-Learn Project. Retrieved July 31st, 2006

[50] Van der Vegt, W., & Koper, R. (2006). CopperAuthor 1.6. Retrieved October 11th, 2006, from www.copperauthor.org

[51] IMSLD. (2003). IMS Learning Design Specification. Retrieved February 27th, 2007, from http://www.imsglobal.org/learningdesign/index.cfm

[52] Koper, R., & Tattersall, C. (2005). *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*. Berlin Heidelberg: Springer.

[53] Burgos, D., & Griffiths, D. (2005). The UNFOLD Project. Understanding and using Learning Design. from http://hdl.handle.net/1820/548

[54] Tattersall, C., & Koper, R. (2003). EML and IMS Learning Design: from LO to LA. Retrieved May 31st, 2005, from http://www.ltsn.ac.uk/application.asp?app=resources.asp&process=full_record&section=generic&id=255,

[55] Koper, R., & Manderveld, J. M. (2004). Educational modelling language: modelling reusable, interoperable, rich and personalised units of learning. *British Journal of Educational Technology, 35*(5), 537-551.

[56] Brusilovsky, P., & Paylo, C. (2003). Adaptive and Intelligent Web-based Educational Systems. *International Journal of Artificial Intelligence in Education, 13*, 156-169.

[57] Brusilovsky, P., Specht, M., & Weber, G. (1995). Towards adaptive learning environments. In F. Huber-Wäschle, H. Schauer & P. Widmayer (Eds.), *Herausforderungen eines globalen Informationsverbundes für die Informatik: GISI 95* (pp. 322-329). Berlin: Springer-Verlag.

[58] Burgos, D., Tattersall, C., & Koper, R. (2005). Utilización de estándares en el aprendizaje virtual*. Proceedings of II Jornadas Campus Virtual*, Madrid: Universidad Complutense de Madrid

[59] LN4LD2. (2005). Units of Learning developed by several authors at Learning Network for Learning Design of The Open University of The Netherlands. Retrieved March 6th, 2006, from http://moodle.learningnetworks.org/course/view.php?id=20 and http://dspace.ou.nl

[60] OUNL. (2002). DSpace. from http://dspace.learningnetworks.org

[61] Koper, R., & Burgos, D. (2005). Developing advanced units of learning using IMS Learning Design level B. *International Journal on Advanced Technology for Learning, 2*(4), 252-259.

[62] Burgos, D., & Corbalan, G. (2006). Modelado y uso de escenarios de aprendizaje en entornos b-learning desde la práctica educativa*. Proceedings of III Jornadas Campus Virtual*. September 26th, 2006, Madrid: Universidad Complutense

[63] Cole, M., & Engestrom, Y. (1993). A cultural-historical approach to distributed cognition. In G. Salomon (Ed.), *Distributed cognitions*. New York: Cambridge University Press.

[64] Shuell, T. J. (1988). The role of the student in learning from instruction. *Contemporary Educational Psychology, 13*, 276-295.

[65] IMSeP. (2005). IMS ePortfolio. 2005, from http://www.imsglobal.org/

[66] Burgos, D., Berbegal, N., Griffiths, D., Tattersall, C., & Koper, R. (2005). *IMS Learning Design: How the specifications can change the current e-learning landscape* (Vol. 2). Moscow: Magazine of Moscow State University for Economy, Statistics and Computer Science.

[67] Burgos, D., Berbegal, N., Griffiths, D., Tattersall, C., & Koper, R. (2005). IMS Learning Design desde dentro. Una especificación para crear escenarios de aprendizaje online. *Learning Review*(9, October-November, 2005).

[68] Burgos, D., & Koper, R. (2005). Practical pedagogical uses of IMS Learning Design's Level B. *Proceedings of SIGOSSEE 2005 Conference*, Heerlen, the Netherlands

[69] Milligan, C. D., Beauvoir, P., & Sharples, P. (2005). The Reload Learning Design Tools. *Journal of Interactive Media in Education (Advances in Learning Design Special Issue), 2005/14*.

[70] Miao, Y. (2005). Cosmos LD Editor. Retrieved October 10th, 2006

[71] European-Comission. (2007). Framework Programme VII. Retrieved March, 1st, 2007, from http://cordis.europa.eu/fp7

[72] Tattersall, C., Burgos, D., Vogten, H., Martens, H., & R. Koper. (2005). How to use IMS Learning Design and SCORM 2004 together*. Proceedings of SCORM 2006 conference*, Taipei, Taiwan [http://ia.nknu.edu.tw/scorm2006/]

[73] Friesen, N. (2005). Interoperability and Learning Objects: An Overview of E-Learning Standardization. *Interdisciplinary Journal of Knowledge and Learning Objects, 1*(1), 23-31.

[74] Mason, J. (2005). From e-learning to e-knowledge. In M. Rao (Ed.), *Knowledge Management Tools and Techniques* (pp. 320-328). Elsevier: London.

[75] Liber, O., & Olivier, B. (2003). Learning Technology Interoperability Standards. In A. Littlejohn & S. Buckingham Shum (Eds.), *Reusing Online Resources: A Sustainable Approach to eLearning*. London: Kogan Page.

[76] Sloep, P. (2002). Learning Technology Standardization. Retrieved May 30th, 2005, from http://eml.ou.nl/introduction/docs/Learning%20Technology%20standardization.pdf

[77] Milligan, C., Gordon, J., & Christie, B. (2002). eLearning Standards Report A resource for the EUROLEARN.net. Retrieved May 30th, 2005, from http://www.eurolearn.net/docs/ELU-standards-02.pdf

[78] MASIE. (2003). Making Sense of Learning Specifications & Standards: A Decision Maker's Guide to their Adoption (2nd Edition). Retrieved May 30th, 2005, from http://www.masie.com/standards/s3_2nd_edition.pdf

[79] Sun. (2002). e-Learning Interoperability Standards. Retrieved May 30th, 2005, from www.sun.com/products-n-solutions/edu/whitepapers/pdf/eLearning_Interoperability_Standards_wp.pdf

[80] Loidl Reisinger S., & Paramythis A. (2003). Distance Education. A Battlefield for standards*. Proceedings of EDEN (European Distance Education Network), The Quality Dialogue. Integrating Quality Cultures in Flexible, Distance and eLearning*, Rhodes, Greece: EDEN

[81] IMSLD. (2003). IMS Learning Design Specification. Version 1. Retrieved February 27th, 2004, from http://www.imsglobal.org/learningdesign/index.cfm

[82] Stiles, M. J. (2003). How does content standardisation impact on staff support in the use of VLEs? *Proceedings of LTSN Supporting Sustainable e-Learning event*, University of York

[83] Ellaway, R., Dewhurst, D., & Cromar, S. (2004). Challenging the Mortality of Computer Assisted Learning Materials in the Life Sciences: The RECAL Project. *Bioscience Education Electronic journal (BEE-j), 3*.

[84] Olivier, B., & Tattersall, C. (2005). The Learning Design Specification. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 21-40): Springer Verlag.

[85] Koper, R., & Olivier, B. (2004). Representing the Learning Design of Units of Learning. *Educational Technology & Society, 7*(3), 97-111.

[86] IMSCP. (2003). IMS Content Packaging Specification. Version 1.1.3. Retrieved February 27th, 2004, from http://www.imsglobal.org/learningdesign/index.cfm

[87] Vogten, H., Tattersall, C., Koper, R., Van Rosmalen, P., Brouns, F., Van Bruggen, J., et al. (forthcoming). Designing a learning design engine as a collection of finite state machines. *International Journal on E-Learning*.

[88] Mackenzie, G. (2004). SCORM 2004 Primer: A (Mostly) Painless Introduction to SCORM. Retrieved May 30th, 2005, from

http://www.mcgill.com/media/SCORM_2004_Primer_v1_McGill_Digital_Solutions_Gord_Mackenzie.pdf

[89] Ostyn, C. (2002, April 17, 2002). Cooking up a SCORM — A SCORM 1.2 Content Cookbook for Developers. Retrieved May 30th, 2005, from http://academiaelearning.com/contenido/scorm/cooking/i_cookbook.htm

[90] Karampiperis, P., & Sampson, D. (2005). Designing Learning Services: From Content-based to Activity-based Learning Systems. *Proceedings of Proc. of the fourteenth International World Wide Web Conference (WWW2005)*, Chiba, Japan

[91] Kraan, W., & Wilson, S. (2002). Dan Rehak: "SCORM is not for everyone". Retrieved May 30th, 2005, from http://www.cetis.ac.uk/content/20021002000737

[92] Wirski, R., Brownfield, G., & Oliver, R. (2004). Exploring SCORM and the national flexible learning toolboxes. *Proceedings of Proceedings of the 21st ASCILITE Conference*, Perth

[93] Neumann, F., & Geys, R. (2004). SCORM and the Learning Grid. *Proceedings of 7th HLRS MetaComputing and Grid Workshop*, Stuttgart, Germany

[94] Beauvoir, P., & Sharples, P. (2005). Reload Learning Design editor.

[95] Dougiamas, M. (2004). Moodle. Retrieved December 12th, 2004, from http://moodle.org/

[96] Wisher, R. A., & Fletcher, J. D. (2004). The Case for Advanced Distributed Learning. *Information & Security: An International Journal, 14*, 17-25.

[97] Sharples, P., & Beauvoir, P. (2005). Reload Learning Design player.

[98] Ip, A., & Canale, R. (2002). Single Copy Re-use of Sharable Content Objects. Retrieved May 26th, 2005, from http://koala.dls.au.com/scorm/

[99] Roberts, E., Freeman, M. W., Wiley, W., & Sampson, D. (2005). Special Issue on SCORM 2004 Sequencing & Navigation. *Educational Technology & Society, 7*(1).

[100] Berggren, A., Burgos, D., Fontana, J.M., Hinkelman, D., Hung, V., Hursh, A. & G. Tielemans. (2005). Practical and Pedagogical Issues for Teacher Adoption of IMS Learning Design Standards in Moodle LMS. *Journal of Interactive Media*(Special issue on Learning Design).

[101] Moodle. (2005). Learning Design book study. Retrieved June 15th, 2007, 2007, from http://moodle.org/course/view.php?id=44

[102] Krueger, R., & Casey, M. (2000). *Focus groups: A practical guide for applied research* (3rd. ed.). Thousand Oaks, CA: Sage.

[103] Morgan, D. (1996). *Focus groups as qualitative research*. Thousand Oaks, CA: Sage.

[104] Jonassen, D., & Land, S. (Eds.). (2000). *Theoretical foundations of learning environments*. Mahweh, NJ: Lawrence Erlbaum.

[105] Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

[106] Turkle, S., & Papert, S. (1992). Epistemological pluralism and the revaluation of the concrete. Retrieved May 25th, 2005, from http://www.papert.org/articles/EpistemologicalPluralism.html

[107] McAndrew, P., & Weller, M. (2005). Applying Learning Design to Supported Open Learning. In E. J. R. Koper & C. Tattersall (Eds.), *Learning Design...* (Vol. 17, pp. 281-290). Berlin: Springer-Verlag.

[108] Suite, e. L. (2005). elive Learning Design. Retrieved May 30th, 2005, 2005, from http://www.elive-ld.com/content/index_eng.html

[109] E., G., & Takayama, Y. (2005). LAMS Foundation: IMS Learning Design. Retrieved December 24th, 2006, 2006, from http://users.ox.ac.uk/~jiscpub/JISC_TIP/documents/LAMS%20-%20IMS%20LD.doc

[110] Lévi-Strauss, C. (1962). *The Savage Mind*. Chicago, IL: University of Chicago Press.

[111] Law, J. (2004). *After method: Mess in social science research*. London: Routledge.

[112] Halm, M., Olivier, B., Farooq, U., & Hoadley, C. (2005). Collaboration in Learning Design Using Peer-to-Peer Technologies. In E. J. R. Koper & C. Tattersall (Eds.), *Learning Design...* (Vol. 11, pp. 203-213). Berlin: Springer-Verlag.

[113] Burgos, D., Tattersall, C., Dougiamas, M., Vogten, H., & Koper, E. J. R. (2006). Mapping IMS Learning Design and Moodle. A first understanding. *Proceedings of Simposo Internacional de Informática Educativa (SIIE06)*, León, Spain: IEEE Technical Committee on Learning Technology. Retrieved September 12th, 2006, from http://dspace.learningnetworks.org

[114] IMS. (2003). IMS Learning Design. Retrieved April 13th, 2006, from www.imsglobal.org

[115] Kenny, R. F., Zhang, Z., Schwier, R. A., & Campbell, K. (2005). A Review of What Instructional Designers Do: Questions Answered and Questions Not Asked *Canadian Journal of Learning and Technology, 31*(1).

[116] Koper, R. (2005). An Introduction to Learning Design. In R. Koper & C. Tattersall (Eds.), *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training* (pp. 3-20). Heidelberg: Springer Verlag.

[117] Van der Vegt, W. (2005). CopperAuthor. Retrieved April 13th 2006, from www.copperauthor.org

[118] Reload. (2004). The Reload Project. Retrieved February 9th 2006, 2006, from http://www.reload.ac.uk

[119] Vogten, H., & Martens, H. (2005). CopperCore 3.0. Retrieved April 13th 2006, from www.coppercore.org

[120] Burgos, D., Arnaud, M., Neuhauser, P. & R. Koper. (2005). IMS Learning Design : la flexibilité pédagogique au service des besoins de la e-formation. *La Revue de l'EPI*.

[121] Huizinga, J. (1971). *Homo Ludens*. UK: Beacon Press.

[122] Caillois, R. (1958). *Les jeux et les homes: Le masque et le vertige*. Paris: Gallimard.

[123] Molnar, A. R. (1990). Computers in Education: A Historical Perspective of the Unfinished Task. *T.H.E. Journal, 18*(4), 80-83.

[124] Molnar, A. R. (1997). Computers in Education: A Brief History. *The journal* Retrieved September 27th, 2006, from http://www.thejournal.com/articles/13739_1

[125] Impagliazzo, J., & Lee, J. A. N. (Eds.). (2004). *History of Computing in Education*. Germany: Springer.

[126] Bruckman, A. (1993). Community support for constructivist learning. *Computer Supported Cooperative Work, 7*, 47-86.

[127] Prensky, M. (2001). *Digital Game-based Learning*. New York, USA: McGraw-Hill.

[128] Dempsey, J. V., Haynes, L. L., Lucassen, B. A., & Casey, M. S. (2002). Forty simple computer games and what they could mean to educators. *Simulation & Gaming, 33*(2), 157-168.

[129] Pivec, M., Dziabenkod, O., & Schinnerl, I. (2003). Aspects of game-based learning. *Proceedings of I-KNOW '03* (pp. 216-225)

[130] Gee, J. P. (2005). Would a State of the Art Instructional Video Game Look Like? *Innovate Journal of online education, 1*(6).

[131] Burgos, D., Tattersall, C., & Koper, R. (2006). Re-purposing existing generic games and simultations for e-learning. *Computers in Human Behavior. Special issue on Education and pedagogy with Learning objects and Learning designs*.

[132] Cobb, P., Stephan, M., McClain, K., & Gravemeijer, K. (2001). Participating in classroom mathematical practices. *Journal of the Learning Sciences, 10*, 113–163.

[133] Squire, K. (2005). Changing the Game: What Happens When Video Games Enter the Classroom? *Innovate Journal of online education, 1*(6).

[134] Jenkins, H., & Squire, K. (2003). Understanding Civilization (III). *Games Magazine, September*.

[135] Eskelinen, M. (2001). The Gaming Situation. *Game Studies, 1*(1).

[136] Williamson, D., Squire, K., Halverson, R., & Gee, J. P. (2005). Video games and the future of learning. *WCER Working paper, 2005-4*.

[137] Squire, K. (2002). Video games in education. *International Journal of Intelligent Simulations and Gaming, 2*(1).

[138] SEGA. (2005). Rome. Total War. Barbarian Invasion. from www.sega.com

[139] Microsoft. (2006). Flight Simulator X. from http://www.microsoft.com/games/flightsimulatorx/

[140] Interactive, E. (2004). Athens 2004.

[141] Microsoft. (2006). Bicycle games board. from http://www.microsoft.com/games/bicycle/board/

[142] OUP (Ed.). (2004). *Caminatas*. Madrid: OUP.

[143] Arts, E. (2005). The Sims 2. Retrieved August 20th, 2006, from www.eagames.com

[144] Auralog. (2005). Talk to me. Retrieved August 20th, 2006, from http://www.auralog.com/es/particulares_talktome.htm

[145] Arts, E. (2005). SimCity 2004. Retrieved August 20th, 2006, from http://simcity.ea.com

[146] Turkle, S. (1995). *Life on the screen: Identity in the age of the internet*. New York: Simon&Schuster.

[147] Piaget, J. (1962). *Play, dreams and imitation in childhood*. New York, USA: Norton.

[148] Vygotksy, L. (1978). *Mind and society*. Cambridge, MA: MIT Press.

[149] Burgos, D. (2005). IMS LD Unit of Learning Caminatas. Retrieved Dec 18th, 2006

[150] Burgos, D., Tattersall, C., & Koper, R. (2006). Re-purposing existing generic games and simultations for e-learning. *Computers in Human Behavior*.

[151] Leyland, B. (1996). How can computer games offer deep learning and still be fun. *Proceedings of Ascilite*. December 2nd-4th, 1996. Retrieved February 4th, 2007

[152] Provenzo, E. F. (1991). *Video kids: making sense of Nintendo*. Cambridge, MA: Harvard University Press.

[153] Dickey, M. D. (2005). Engaging by design: How engagement strategies in popular computer and video games can inform instructional design. *Educational Technology Research & Development, 53*(2), 67-83.

[154] Squire, K., & Barab, S. (2004). Replaying History: Engaging Urban Underserved Students in Learning World History Through Computer Simulation Games. In Y. B. Kafai, W. A. Sandoval, N. Enyedy, A. S. Nixon & F. Herrera (Eds.), *Embracing diversity in the Learning Sciences: Proceedings of the sixth international conference of the Learning Sciences* (pp. 505-512). Mahwah, NJ: Lawrence Erlbaum Associates.

[155] Crawford, C. (1984). *The art of computer game design*. New York: McGraw-Hill.

[156] Arts, E. (2005). SimCity 2004. Retrieved Feb 14th, 2008, from http://simcity.ea.com

[157] Zipkin, G. (2004). Tetris 3. Retrieved November 16th, 2006, from http://www.csc.calpoly.edu/~zwood/teaching/csc471/finalproj24/gzipkin

[158] Richards, G. (2005). Designing Educational Games. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. Chapter 13). Germany: Springer Verlag.

[159] LTd, H. A. (2004). Trivial Pursuit. Retrieved November 16th, 2006, from http://www.trivialpursuit.com

[160] Arts, E. (2005). The Sims 2. Retrieved Feb 14th, 2008, from www.eagames.com

[161] Burgos, D., Moreno-Ger, P., Sierra, J. L., Fernández-Manjón, B., & Koper, R. (in press). Authoring Game-Based Adaptive Units of Learning with IMS Learning Design and <e-Adventure>. *International Journal on Learning Technologies*.

[162] Sierra, J. L., Fernández-Manjón, B., Fernández-Valmayor, A., & Navarro, A. (2005). Document Oriented Development of Content-Intensive Applications. *International Journal of Software Engineering and Knowledge Engineering, 15*(6), 975-993.

[163] Sierra, J. L., Fernández-Valmayor, A., Fernández-Manjón, B., & Navarro, A. (2004). ADDS: A Document-Oriented Approach for Application Development. *Journal of Universal Computer Science, 10*(9), 1302-1324.

[164] Sierra, J. L., Navarro, A., Fernández-Manjón, B., & Fernández-Valmayor, A. (2005). Incremental Definition and Operationalization of Domain-Specific Markup Languages in ADDS. *ACM SIGPLAN Notices, 40*(12), 28-37.

[165] Martinez-Ortiz, I., Moreno-Ger, P., Sierra, J. L., & Fernández-Manjón, B. (2006). Production and Deployment of Educational Videogames as Assessable Learning Objects*. Proceedings of First European Conference on Technology Enhanced Learning (ECTEL 2006)* (pp. 316-330), Crete, Greece: Lecture Notes in Computer Science, Springer

[166] Moreno-Ger, P., Martínez-Ortiz, I., Sierra, J. L., & Fernández-Manjón, B. (2006). Language-Driven Development of Videogames: The <e-Game> Experience*. Proceedings of 5th International Conference in Entertainment Computing (ICEC 2006)* (pp. 153-164). 20-22 September 2006, Cambridge, UK: Lecture Notes in Computer Science, Springer

[167] Moreno-Ger, P., Sierra, J. L., Martínez-Ortiz, I., & Fernández-Manjón, B. (In press). A Documental Approach to Adventure Game Development. *Science of Computer Programming, In press*.

[168] Bray, T., Paoli, J., Sperberg-McQueen, C. M., & Maler, E. (2000). Extensible Markup Language (XML) 1.0. *W3C Recommendation* Retrieved July 24th, 2006, from www.w3c.org

[169] Lee, D., & Chu, W. W. (2000). Comparative Analysis of Six XML Schema Languages. *ACM SIGMOD Record, 29*(3), 76-87.

[170] Murata, M., Lee, D., Mani, M., & Kawaguchi, K. (2005). Taxonomy of XML schema languages using formal language theory. *ACM Transactions on Internet Technology, 5*(4), 660-704.

[171] LN4LD. (2005). Units of Learning developed by several authors at Learning Network for Learning Design of The Open University of The Netherlands. Retrieved March 6th, 2006, from http://moodle.learningnetworks.org/course/view.php?id=20 and http://dspace.ou.nl

[172] Goldfarb, C. F. (1981). A Generalized Approach to Document Markup. *ACM SIGPLAN Notices, 16*(6), 68-73.

[173] Ahmad, A., Basir, O., & Hassanein, K. (2004). Adaptive user interfaces for intelligent e-Learning: issues and trends*. Proceedings of The Fourth International Conference on Electronic Business, ICEB2004*, Beijing

[174] Cristea, A. (2005). Authoring of Adaptive Hypermedia. *Educational Technology & Society, 8*(3), 6-8.

[175] Chen, S. Y., & Magoulas, G. D. (2005). *Adaptable and Adaptive Hypermedia Systems*. Hershey, PA: IRM Press.

[176] Burgos, D., Tattersall, T., & Koper, R. (2006). Representing adaptive eLearning strategies in IMS Learning Design*. Proceedings of TENCompetence Conference*. March 31st, 2006, Sofia, Bulgaria

[177] Klann, M. (2003). The EUD-Net's Roadmap to End-User Development*. Proceedings of Workshop on End User Development at ACM CHI 2003 Conference*. April 6, 2003, Fort Lauderdale, USA

[178] Tennyson, R. D. (1980). Instructional control strategies and content structure as design vairables in concept acquisition using computer-based instruction. *Journal of Educational Psychology, 72*, 525-532.

[179] Tennyson, R. D. (1981). Use of adaptive information for advisement in learning concepts and rules using computer assisted instruction. *American Educational Research Journal, 18*, 425-438.

[180] Bell, B., & Kozlowski, S. (2002). Adaptive guidance: Enhancing self-regulation, knowledge, and performance in technology-based training. *Personnel Psychology, 55*(2), 267-306.

[181] Van Rosmalen, P., Vogten, H., Van Es, R., Van, P., H., Poelmans, P., & Koper, R. (2006). Authoring a full life cycle model in standards-based, adaptive e-learning. *Educational Technology & Society, 9*(1), 72-83.

[182] Wasson, B. (1997). Advanced educational technologies: The learning environment. *Computers in Human Behavior, 13*(4), 571-594.

[183] He, S., Kinshuk, Hong H., & Patel A. (2002). Granular Approach to Adaptivity in Problem-based Learning Environment*. Proceedings of ICALT* (pp. 3-7)

[184] Fredericksen, E., Pickett, A., Shea, P., & Pelz, W. (2000). Student satisfaction and perceived learning with on-line courses: Principles and examples from the SUNY Learning Network. *Journal of Asynchronous Learning Networks, 4*(2).

[185] Burgos, D., & Ruiz-Mezcua, B. (2003). Building an interactive training methodology to develop multimedia elearning software*. Proceedings of International Conference On Education, Information Systems, Technologies and Applications EISTA 2003*, Orlando, Florida, USA

[186] Towle, B., & Halm, M. (2005). Designing adaptive learning environments with Learning Design. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modeling and Delivering Networked Education and Training.* Heidelberg, Germany: Springer Verlag.

[187] Snow, R. (1980). Aptitude, learner control and adaptive instruction. *Educational Pshychologist, 15*(13), 151-158.

[188] De Bra, P., Stash, N., & Smits, D. (2005). Creating Adaptive Web-Based Applications*. Proceedings of 10th International Conference on User Modeling*. July 25, 2005, Edinburgh, Scotland

[189] Mödritscher, F., García, V., & Gütl, C. (2004). The Past, the Present and the Future of adaptive E-Learning. An Approach within the Scope of the Research Project AdeLE*. Proceedings of ICL*, Villach, Austria

[190] Park, O., & Lee, J. (2003). Adaptive Instructional Systems. *Educational Technology Research and Development, 25*, 651-684.

[191] Burgos, D., Nimwegen, C. v., Oostendorp, H. v., & Koper, E. J. R. (2006). Game-based learning and inmediate feedback. The case study of the Planning Educational Task. *International Journal of Advanced Technology in Learning*.

[192] Mory, E. H. (1996). Feedback Research Revisited. In D. H. Jonassen (Ed.), *Handbook of*

research for educational communications and technology: A Project of the Association for Educational Communications and Technology (pp. 745-774). New York, USA: Macmillan.

[193] Garries, R., Ahlers, R., & Driskel, J. E. (2002). Game based learning cycle. *Simulation & Gaming, 33*(4), 441-467.

[194] Annett, J. (1969). *Feedback and human behavior*. Middlesex, UK: Penguin.

[195] Csikszentmihalyi, M. (1990). *Flow, the psychology of optimal experience*. USA: Harper & Row.

[196] Kernan, M. C., & Lord, R. G. (1990). Effects of valence, expectancies, and goal-performance discrepancies in single and multiple goal environments. *Journal of Applied Psychology, 75*, 194-203.

[197] Beck, J. C., & Wade, M. (2004). *Got Game: how the gamer generation is reshaping business forever*. Boston, MA: Harvard Business School Press.

[198] Baer, L. L. (2005). The Generation Gap: Bridging Learners and Educators. *The International Digital Media & Arts Association Journal, 2*(1).

[199] Kirriemuir, J. (2002). *The relevance of video games and gaming consoles to the higher and further education learning experience*. UK: Techwatch Report.

[200] Kiili, K. (2005). Digital game-based learning: Towards and experiential gaming model. *Internet and Higher Education, 8*, 13-24.

[201] Chen, H., Wigand, R., & Nilan, M. S. (1999). Optimal experience of web activities. *Computers in Human Behavior, 15*, 585-608.

[202] Mitchell, A., & Savill-Smith, C. (2004). *The use of computer and video games for learning. A review of the literature*. UK: Learning and Skills Development Agency.

[203] Halttunen, K., & Sormunen, E. (2000). Learning information retrieval through an educational game. Is gaming sufficient for learning? *Education for Information, 18*(4), 289–311.

[204] Fowlkes, J., Dwyer, D. J., Oser, R. L., & Salas, E. (1998). Event-based approach to training (EBAT). *The International Journal of Aviation Psychology, 8*(3), 209-222.

[205] Kluger, A. N., & DeNisi, A. (1996). The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological Bulletin, 119*, 254-284.

[206] Mason, B. J., & Bruning, R. (1999). Providing Feedback in Computer-based Instruction: What the Research Tells Us. Retrieved November 29th, 2006, from http://dwb.unl.edu/Edit/MB/MasonBruning.html

[207] Evans, K., Kersh, N., & Kontianen, S. (2004). Recognition of tacit skills: sustaining learning outcomes in adult learning and work re-entry. *International Journal of Training and Development, 8*(1), 54-72.

[208] Gherardi, S. (2006). *Organizational knowledge, the texture of workplace learning*. Malden, MA: Blackwell.

[209] Computer, A. (1992). *Macintosh Human Interface Guidelines*: Apple Computer, Inc.

[210] Gentner, D., & Nielsen, J. (1996). The Anti-Mac Interface. *Communications of the ACM, 39*(8), 70-82.

[211] Janssen, J., Erkens, G., Jaspers, J., & Broeken, M. (2005). *Effects of visualizing participation in computer-supported collaborative learning.* Paper presented at the conference First ICO annual research meeting. November, Utrecht, The Netherlands

[212] Weber, R. A. (2003). *Learning and Transfer of Learning with No Feedback: An Experimental Test Across Games*.

[213] Zhang, J. (1997). The nature of external representations in problem solving. *Cognitive Science, 21*(2), 179-217.

[214] Zhang, J., & Norman, D. A. (1994). Representations in distributed cognitive tasks. *Cognitive Science, 18*, 87-122.

[215] O'Hara, K. P., & Payne, S. J. (1999). Planning and the user interface: The effects of lockout time and error recovery cost. *International Journal of Human-Computer Studies, 50*(1), 41-59.

[216] Svendsen, G. B. (1991). The influence of interface style on problem solving. *International Journal for Man-Machine Studies, 35*.

[217] Cronbach, L. J. (1957). The two disciplines of scientific psychology. *American Psychologist, 12*, 671-684.

[218] Chin, D. (2001). Empirical Evaluation of User Models and User-Adapted Systems. *User Modeling and User-Adapted Interaction, 11*, 181-194.

[219] Brusilovsky, P., & Miller, P. (2001). Course Delivery Systems for the Virtual University. In Tschang F.T. & T. Della Senta (Eds.), *Access to Knowledge: New Information Technologies and the Emergence of the Virtual University* (pp. 167-206). Amsterdam: Elsevier Science and International Association of Universities.

[220] De Bra, P., Aroyo, L., & Cristea, A. (2004). Adaptive Web-based Educational Hypermedia. In M. Levene & A. Poulovassilis (Eds.), *Web Dynamics, Adaptive to Change in Content, Size, Topology and Use* (pp. 387-410): Springer.

[221] Baeza-Yates, R., & Ribeiro-Nieto, B. (1999). *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley.

[222] Van Rosmalen, P., & Boticario, J. (2005). Using Learning Design to support design- and runtime adaptation. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modeling and Delivering Networked Education and Training*. Heidelberg, Germany: Springer Verlag.

[223] Merceron, A., & Yacef, K. (2003). A Web-based tutoring tool with mining facilities to improve learning and teaching. *Proceedings of AI-Ed'2003*: IOS Press

[224] Romero, C., Ventura, S., De Bra, P. D., & Castro, C. D. (2003). Discovering prediction rules in AHA! courses. *Proceedings of 9th International User Modeling Conference* (pp. 25-34)

[225] Specht, M., & Burgos, D. (2006). Implementing Adaptive Educational Methods with IMS Learning Design. *Proceedings of Adaptive Hypermedia 2006*, Dublin, Ireland

[226] Butz, M. V., Olivier, S., & Gérard, P. (2003). Anticipatory Behavior in Adaptive Learning Systems : Foundations, Theories, and Systems. In. Berlin: Springer Verlag.

[227] Koper, R., & Tattersall, C. (Eds.). (2005). *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training*. Heidelberg: Springer Verlag.

[228] Vogten, H., & Martens, H. (2005). CopperCore. Retrieved February 9th, 2006, from www.coppercore.org

[229] Reload. (2004). The Reload Project. Retrieved February 9th, 2006, from http://www.reload.ac.uk/editor.html

[230] The Open University. (2005). The Sled player. Retrieved February 9th, 2006, from http://sled.open.ac.uk

[231] Koper, R., & Burgos, D. (2005). Developing advanced units of learning using IMS Learning Design level B. *International Journal on Advanced Technology for Learning, 2*(3).

[232] Van der Vegt, W. (2005). CopperAuthor. Retrieved February 23rd, 2005, from www.copperauthor.org

[233] Burgos, D. (2004). Step by step. How to install CopperCore, how to publish and run a UoL. Retrieved February 21st, 2005, from http://hdl.handle.net/1820/327

[234] Vogten, H., Martens, H., Nadolski, R., Tattersall, C., Van Rosmalen, P., & Koper, E. J. R. (2006). Integrating IMS Learning Design and IMS Question and Test Interoperability using CopperCore Service Integration. *Proceedings of Proceedings of the International Workshop in Learning Networks for Lifelong Competence Development at the TENCompetence Conference in March 30th-31st, 2006*, Sofia, Bulgaria: TENCompetence

[235] ALBARELLO, L. (1995). *Pratiques et méthodes de recherche en Sciences Sociales*. Paris: Armand Colin.

[236] BRYMAN, A., & BURGESS, R. (Eds.). (1994). *Analyzing qualitative data*. London: Routledge.

[237] García Ferrando, M., Ibáñez, J., & Alvira, F. (Eds.). (2000). *El análisis de la realidad social: metodología y técnicas de investigación*. Madrid: Alianza Editorial.

[238] Hansen, P. B. (2002). *The Origin of Concurrent Programming: From Semaphores to Remote Procedure Calls*: Springer.

[239] Snow, C. R. (1992). *Concurrent programming*: Cambridge University Press.

[240] Taubenfeld, G. (2006). *Synchronization Algorithms and Concurrent Programming*: Pearson, Prentice Hall.

[241] Kastens, U., Waite, W. M., & Sloane, A. M. (2007). *Generating Software from Specifications*: Jones & Bartlett Publishers.

[242] Knuth, D. E. (1992). Literate Programming. *Proceedings of CSLI*, Stanford, California: Center for the Study of Language and Information

[243] Tattersall, C., Sodhi, T., Burgos, D., & Koper, R. (in press). Using the IMS Learning Design notation for the modelling and delivery of education. In L. Lockyer, S. Bennet, S. Agostinho & B. Harper (Eds.), *Handbook of Research on Learning Design and Learning Objects: Issues, Applications and Technologies*. Wollongong: Idea Group Inc.

[244] Merrill, M. D., & Wilson, B. (2006). The Future of Instructional Design (Point/Counterpoint). In R. Reiser & J. V. Dempsey (Eds.), *Trends and Issues in Instructional Design and Technology, Second Edition* (pp. 335-351). New Jersey, USA: Prentice Hall.

[245] Kirschner, P., Carr, C., Van Merriënboer, J. J. G., & Sloep, P. (2002). How Expert Designers Design. *Performance Improvement Quarterly, 15*(4), 86-104.

[246] Eseryel, D., Schuver-van Blanken, M., & Spector, J. M. (2001). Current practice in designing training for complex skills: Implications for design and evaluation of ADAPT-IT. In C. Montgomerie & J. Vitelli (Eds.), *Proceedings of ED-MEDIA 2001: World Conference on Educational Multimedia, Hypermedia, & Telecommunications* (pp. 474-479). Tampere , Finland Association for Advancement of Computing in Education.

[247] Waters, S. H., & Gibbons, A. S. (2004). Design Languages, Notation Systems and Instrictional Technology: A Case Study. *Education Technology Research & Development, 52*(2), 57-63.

[248] Tattersall, C., Vogten, H., Brouns, F., Koper, R., Van Rosmalen, P., Sloep, P., et al. (2005). How to create flexible runtime delivery of distance learning courses. *Educational Technology & Society, 8*(3), 226-236.

[249] Gibbons, A. S., & Brewer, E. K. (2005). Elementary Principles of Design Languages and Design Notation Systems for Instructional Design. In J. M. Spector, C. Ohrazda, A. Van Schaak & D. A. Wiley (Eds.), *Innovations in Instructional Technology: Essays in honor of M. David Merrill* (pp. 111-130). Mahwah, New Jersey: Lawrence Erlbaum Associates.

[250] Süß, C., & Freitag, B. (2002). LMML - The Learning Material Markup Language Framework*. Proceedings of International Workshop: Interactive Computer Aided Learning*, Villach, Austria,

[251] Rawlings, A., Van Rosmalen, P., Koper, R., Rodríguez-Artacho, M., & Lefrere, P. (2002). *CEN/ISSS WS/LT Learning Technologies Workshop: "Survey of Educational Modelling Languages (EMLs)"*.

[252] Koper, R. (2005). *Modeling lifelong learning networks.* Paper presented at the conference IPSI Conference. April 29-30, Spain

[253] Burgos, D., & Griffiths, D. (2005). *The UNFOLD Project. Understanding and Using Learning Design*. Heerlen: Open University of The Netherlands.

[254] ISO. (2005). *ISO/IEC 19501:2005 Information technology -- Open Distributed Processing -- Unified Modeling Language (UML) Version 1.4.2* Geneva, Switzerland International Organization for Standardization.

[255] Van Es, R., & Koper, R. (2006). Testing the pedagogical expressiveness of IMS LD. *Educational Technology & Society, 9*(1), 229-249.

[256] W3C. (1999). HTML 4.01 Specification, W3C Recommendation. Retrieved October 11th, 2006, from http://www.w3.org/TR/html401/

[257] Koper, R. (2003). Learning technologies: an integrated domain model. In W. Jochems, J. Van Merrienboer & R. Koper (Eds.), *Integrated eLearning* (pp. 64-79). London: RoutledgeFalmer.

[258] Martens, H., & Vogten, H. (2005). A Reference Implementation of a Learning Design Engine. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook*

on Modelling and Delivering Networked Education and Training (pp. 91-108). Berlin-Heidelberg: Springer Verlag.

[259] Vogten, H., Koper, R., Martens, H., & Tattersall, C. (2005). An Architecture for Learning Design Engines. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 75-90). Berlin-Heidelberg: Springer Verlag.

[260] Sloep, P., Hummel, H., & Manderveld, J. M. (2005). Basic Design Procedures for E-learning Courses. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 139-160). Berlin-Heidelberg: Springer Verlag.

[261] Janssen, J., & Hermans, H. (2005). How to Integrate Learning Design into Existing Practice. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 253-266). Berlin-Heidelberg: Springer Verlag.

[262] Barrett Baxendale, M. (2005). E-Learning Technical Framework and Tools – Demonstrator Projects: SLeD Integration Demonstrator Final Report. Retrieved October 11th, 2006, from http://www.hope.ac.uk/slide/documents/slide_final.doc

[263] Miao, Y. (2005). CoSMoS: Facilitating Learning Designers to Author Units of Learning Using IMS LD In C. K. Looi, D. H. Jonassen & I. Mitsuru (Eds.), *Proceediongs of the International Conference on Computers in Education* (pp. 275-282). Singapore: IOS Press.

[264] Yu, D., Zhang, W., & Chen, X. (2006). New Generation of E-Learning Technologies In *Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences - Volume 2 (IMSCCS'06)* (pp. 455-459).

[265] Sampson, D. G., Karampiperis, P., & Zervas, P. (2005). ASK-LDT: A Web-Based Learning Scenarios Authoring Environment Based on IMS Learning Design *Advanced Technology for Learning 2005*(4).

[266] Spang Bovey, N., & Dunand, N. (2006). Seamless production of interoperable e-Learning units: stakes and pitfalls. In R. Koper & K. Stefanov (Eds.), *Proceedings of the Workshop on Learning Networks for Lifelong Competence Development,* . Sofia, Bulgaria: INCOMA Ltd.

[267] Griffiths, D., & Blat, J. (2005). The role of teachers in editing and authoring units of learning using IMS Learning Design. *Advanced Technology for Learning 2005*(4).

[268] Brouns, F., Koper, R., Manderveld, J., Bruggen, J. v., Sloep, P., Rosmalen, P. v., et al. (2005). A first exploration of inductive analysis approach for detecting learning design patterns. *Journal of Interactive Media in Education (Advances in Learning Design Special Issue, eds. Colin Tattersall, Rob Koper)*(03).

[269] Bennett, S., Lockyer, L., & Agostinho, S. (2004). Investigation how learning designs can be used as a framework to incorporate learning objects*. Proceedings of Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference* (pp. 116-122), Perth, Australia

[270] Hernández-Leo, D., Harrer, A., Dodero, J. M., Asension-Pérez, J. I., & Burgos, D. (2006). Creating by reusing Learning Design solutions*. Proceedings of Proceedings of the 8th Simposo Internacional de Informática Educativa*, León, Spain: IEEE Technical Committee on Learning Technology

[271] McAndrew, P., Goodyear, P., & Dalziel, J. (2005). Patterns, designs and activities: unifying descriptions of learning structures [Electronic Version]. Retrieved 13th December 2006 from http://kn.open.ac.uk/public/document.cfm?docid=5295.

[272] Alexander, C. (1979). *The Timeless Way of Building*. New York: Oxford University Press.

[273] Bergin, J., Eckstein, J., Manns, M., Sharp, H., & Voelter, M. (2005). Pedagogical Patterns. Retrieved 13th December 2006, from http://www.pedagogicalpatterns.org

[274] Hernández Leo, D., Asensio Pérez, J. I., & Dimitriadis, Y. A. (2004). IMS Learning Design Support for the Formalization of Collaborative Learning Patterns. *Proceedings of Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04)*. 30 August - 1 September, 2004, Joensuu, Finland: IEEE

[275] Olivier, B. (2004). Discussion of Learning Design, State of Play. *UNFOLD* Retrieved 13th December, 2006, from https://www.unfold-project.net:8082/UNFOLD/about_folder/events/online/billjuly04/

[276] Hernández-Leo, D., Villasclaras-Fernández, E. D., Asensio-Pérez, J. I., Dimitriadis, Y., Jorrín-Abellán, I. M., Ruiz-Requies, I., et al. (2006). COLLAGE: A collaborative Learning Design editor based on patterns. *Educational Technology & Society, 9*(1), 58-71.

[277] Es, R. v., & Koper, R. (2006). Testing the pedagogical expressiveness of IMS LD. *Education Technology & Society, 9*(1), 229-249.

[278] McAndrew, P., & Goodyear, P. (2004). Representing practitioner experiences through learning design and patterns. . from http://www.jisc.ac.uk/uploaded_documents/practioner-patterns-v2.doc

[279] Koper, R., & Burgos, D. (2005). Designing Learning Activities: From Content-based to Context-based Learning Services. *International Journal on Advanced Technology for Learning, 2*(3).

[280] Griffiths, D., & Blat, J. (2005). The role of teachers in editing and authoring units of learning using IMS Learning Design. *Advanced Technology for Learning, 2005*(4).

[281] Rungtusanatham, M., Ellram, L. M., Siferd, S. P., & Salik, S. (2004). Toward a Typology of Business Education in the Internet Age *Decision Sciences The Journal of Innovative Education, 2*(2), 101-120.

[282] Dagada, R., & Jakovljevic, M. (2004). 'Where have all the trainers gone?' E-learning strategies and tools in the corporate training environment. In *Proceedings of the 2004 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries table of contents*. Stellenbosch, Western Cape, South Africa ACM.

[283] Trombley, B. K., & Lee, D. (2002). Web-based Learning in Corporations: who is using it and why, who is not and why not? . *Journal of Educational Media, 27*(3), 137-146.

[284] Arlow, J., Emmerich, W., & Quinn, J. (1999). Literate Modelling Capturing Business Knowledge with the UML. In *Proceedings of The First International Workshop on the Unified Modeling Language, UML'98 - Beyond the Notation*. Mulhouse, France.

[285] Sutcliffe, A., & Mehanddjiev, N. (2004). Special Issue: End-user development - tools that empower users to create their own software solutions. *Communications of the ACM, 47*(9).

[286] Hoogveld, A. W. M. (2003). *The Teacher as Designer of Competency-Based Education.* Open Universiteit Nederland, Heerlen.

[287] Romiszowski, A. J. (1984). *Producing Instructional Systems*. London: Kogan Page.

[288] Dahlgren, M. A., & Öberg, G. (2001). Questioning to learn and learning to question: Structure and function of problem-based learning scenarios in environmental science education. *Higher Education, 41*, 263-282.

[289] Pernin, J. L., A. (2005). Modèles pour la réutilisation des scénarios d'apprentissage : limites d'IMS LD. 2007, from http://www-clips.imag.fr/arcade/User/jean-philippe.pernin/recherche/download/PerninLejeune_UNFOLD2005_English.pdf

[290] Ward, R. (1998). Active, collaborative, and case-based learning with computer-based case scenarios. *Computer Education, 30*, 103-110.

[291] Snoek, M. (2003). Scenarios as a tool for reflection and learning. *European Journal of Teacher Education, 26*(1), 3-7.

[292] Gros, B., Elen, J. Kerres, M. Merriënboer, J. J. G., & Spector, M. (1997). Instructional Design and the Authoring of Multimedia and Hypermedia Systems: Does a Marriage make Sense? *Educational Technology, 37*(1), 48-56.

[293] Conole, G., Dyke, M., Oliver, M., & Seale, J. (2004). Mapping pedagogy and tools for effective learning design. *Computers and Education, 43*, 17-33.

[294] Derntl, M. M.-P., R. (2005). The role of structure, patterns, and people in blended learning. *Internet and Higher Education, 8*(111-130).

[295] Reigeluth, C. M. (1996). A new paradigm of ISD? *Educational Technology, 36*(3), 13-20.

[296] Brusilovsky, P. (2001). Adaptive hypermedia. *User Modelling and User-Adapted Interaction, 11*, 87-110.

[297] Park, O., & Lee, J. (2004). Adaptive instructional systems. In D. H. Jonassen (Ed.), *Handbook of research for educational communications and technology* (2nd ed., pp. 651-684). Mahwah, NJ: Erlbaum.

[298] Burgos, D., & Fernández-Valmayor, A. (2005). UNFOLD/Complutense Workshop , Madrid Dec 12-14, 2005 [http://moodle.learningnetworks.org/course/view.php?id=37]. Retrieved May 9th, 2006, 2006, from http://moodle.learningnetworks.org/course/view.php?id=37

[299] Burgos, D., & Rodríguez, G. (2006). ProLearn Workshop on Adaptive and Collaborative Learning. Cadiz, Spain. March 8th-10th [http://moodle.learningnetworks.org/course/view.php?id=42]. Retrieved May 9th, 2006, 2006

[300] ADL. (2000). Sharable Object Reference Model, SCORM. Retrieved May 9th, 2006, 2006, from http://www.adlnet.org/index.cfm?fuseaction=Scormabt

[301] IMS Global Learning Consortium. (2005). Content Packaging Specification. 2006, from http://www.imsglobal.org/content/packaging/index.html

[302] Burgos, D., Tattersall, C., & Koper, E. J. R. (2006). Representing adaptive and adaptable Units of Learning. How to model personalized eLearning in IMS Learning Design. In B. Fernández Manjon, J. M. Sanchez Perez, J. A. Gómez Pulido, M. A. Vega Rodriguez & J. Bravo (Eds.), *Computers and Education: E-learning - from theory to practice*. Germany: Kluwer.

[303] Specht, M., & Burgos, D. (2006). Implementing Adaptive Educational Methods with IMS Learning Design. *Proceedings of Adaptive Hypermedia 2006*, Dublin (Ireland)

[304] Miao, Y., Burgos, D., Griffiths, D., & Koper, R. (in press). Representation of Coordination Mechanisms in IMS-LD. In L. Lockyer, S. Bennet, S. Agostinho & B.

Harper (Eds.), *Handbook of Research on Learning Design and Learning Objects: Issues, Applications and Technologies*. Wollongong: Idea Group Inc.

[305] Dillenbourg, P. (1999). What do you mean by collaborative learning? In P. Dillenbourg (Ed.), *Collaborative-learning: Cognitive and Computational Approaches* (pp. 1-19): Oxford: Elsevier.

[306] Tribe, D. M. R. (1994). An Overview from Higher Education. In L. Thornley & R. Gregory (Eds.), *Using Group-based Learning in Higher Education* (pp. 25-31). London: Kogan Page Limited.

[307] Strijbos, J. W., & Martens, R. L. (2001). Group-based learning: Dynamic interaction in groups*. Proceedings of EURO-CSCL Conference 2001*. March 22-24, Maastricht, The Netherlands

[308] Strijbos, J. W., Martens, R. L., & Jochems, W. M. G. (2004). Designing for interaction: Six steps to designing computer-supported group-based learning. *Computer & Education, 42*, 403-424.

[309] Johnson, D. W. (1981). Student-student interaction: the neglected variable in education. *Educational Research, 10*, 5-10.

[310] Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL* (pp. 61-91). Heerlen, The Netherlands: Open Universiteit Nederland.

[311] Fischer, F., Kollar, I., Mandl, H., & Haake, J. (2007). *Scripting Computer-Supported Collaborative Learning: Cognitive, Computational and Educational Perspectives*: Springer.

[312] Kollar, I., Fischer, F., & Slotta, J. D. (2005). Internal and external collaboration scripts in web-based science learning at schools. In *Proceedings of Computer Supported Collaborative Learning (CSCL2005): The Next 10 Years*. Mahwah, NJ: Lawrence Erlbaum Associates.

[313] Miao, Y., Hoeksema, K., Hoppe, U., & Harrer, A. (2005). CSCL Scripts: Modelling features and potential use. In *Proceedings of Computer Supported Collaborative Learning (CSCL2005): The Next 10 Years*. Mahwah, NJ: Lawrence Erlbaum Associates.

[314] O'Donnell, A. M., & Dansereau, D. F. (1992). Scripted Cooperation in Student Dyad: A Method for Analyzing and Enhancing Academic Learning and Performance. In R. Hertz-Lazarowitz & N. Miller (Eds.), *Interaction in Cooperative Groups: The theoretical Anatomy of Group Learning* (pp. 120-141). London: Cambridge University Press.

[315] Weinberger, A., Stegmann, K., & Fischer, F. (2005). Computer-supported collaborative learning in higher education: Scripts for argumentative knowledge construction in distributed groups. In *Proceedings of Computer Supported Collaborative Learning (CSCL2005): The Next 10 Years*. Mahwah, NJ: Lawrence Erlbaum Associates.

[316] Caeiro, M., Anido, L., & Llamas, M. (2003). A Critical Analysis of IMS Learning Design*. Proceedings of CSCL* (pp. 363-367)

[317] Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys, 26*(1), 87-119.

[318] Crowston, K., & Osborn, C. (1998). *A coordination theory approach to process description and redesign*: Massachusetts Institute of Technology, Centre for Coordination Science.

[319] Weinberger, A., Fischer, F., & Mandl, H. (2004). Knowledge convergence in computer-mediated learning environments: Effects of collaboration scripts. In

Proceedings of the 85th Annual Meeting of the American Educational Research Association. San Diego, CA, USA: AERA.

[320] Miao, Y. (2007). KCS Unit of Learning. Retrieved July 22, 2007, from http://hdl.handle.net/1820/1003

[321] Nimwegen, C. v., Oostendorp, H. v., Schijf, H. J. M., & Burgos, D. (2005). The Paradox of the Assisted User: Guidance Leads to more Shallow Behavior. *Proceedings of SIGCHI conference on Human factors in computing systems*. April 22 - 27, Montreal, Quebec, Canada: ACM Press

[322] Burgos, D., & van Nimwegen, C. (2005). Planning Educational Task. August 15th, 2006, from http://moodle.learningnetworks.org/course/view.php?id=45

[323] Macromedia/Adobe. (2004). Macromedia Flash. from http://www.adobe.com/products/flash/flashpro/

[324] UNFOLD. (2004). The UNFOLD Project. Retrieved June 5th, 2006, from www.unfold-project.net

[325] Van Nimwegen, C., Van Oostendorp, H., & Tabachneck-Schijf, H. J. M. (2004). Does context sensitivity in the interface help? *Proceedings of European Conference on Cognitive Ergonomics* (pp. 29-33): European Asociation of Cognitive Ergonomics

Appendix: evaluation data

PhD Thesis	*Extension of the IMS Learning Design Specification based on Adaptation and Integration of Units of Learning*
Tesis doctoral	*Extensión de la especificacion IMS Learning Design desde la Adaptación y la Integración de Unidades de Aprendizaje*
Author	*Daniel Burgos (daniel.burgos@atosresearch.eu)*
Date of poll	*25-nov-07*
Feedback enddate	*17-dec-07*

Questionnaire

Please, fill in the yellow-coloured slots only

Personal Information

Name	*RESULTS on Jan 3rd, 2008*
Meassurement	*PERCENTAGE*
Sample	*64*

Research and-or interest field/s (choose 1-n)	
Educational Technology	90,6
Computer Science	26,6
Education and Learning	100,0
Adaptive and Personalized Learning	100,0
Interoperability and Systems Integration	26,6
Standards, eLearning Specifications, EMLs	40,6
Others	6,3

Profile/s (choose 1-n)	
Learning Designer	81,3
Teacher	54,7
Content Provider	0,0
Software/System Developer	26,6
System Administrator	14,1
Standards and EMLs Expert	12,5
Others	4,7

Evaluation of the PhD Thesis

Please, mark the choice that fits better with your opinion. Use only the yellow-coloured cells
You can also provide some additional comment either in English or Spanish
Your contribution is highly appreciated and will be useful to strenghen the thesis and the related ongoing research

1. About the topic and the context	Little-1	Sligthly Moderate-2	Moderate-3	Highly Moderate-4	Much-5
1.1 How important is personalization in eLearning?	0,0	0,0	0,0	11,4	88,6
1.2 How important is interoperability in eLearning?	0,0	13,6	0,0	13,6	72,7
1.3 How important are standards in eLearning?	0,0	12,9	14,5	14,5	58,1
1.4 How important are Educational Modelling Languages in eLearning?	0,0	14,5	0,0	41,9	43,5
1.5 How relevant is the research topic of this thesis?	0,0	0,0	11,8	11,8	76,5
1.6 How significant are the topics "adaptive learning" and "interoperability" in eLearning specs?	0,0	0,0	0,0	35,8	64,2
1.7 How relevant is IMS Learning Design among the different specifications and EMLs?	0,0	0,0	13,1	73,8	13,1
2. The methodology used in this thesis to research about the topic is					
Clear	0,0	0,0	13,2	36,8	50,0
Useful	0,0	0,0	13,2	38,2	48,5
Re-usable	0,0	0,0	0,0	73,8	26,2
Original	0,0	0,0	49,3	25,4	25,4
Consistent	0,0	0,0	0,0	63,8	36,2
Efficient	0,0	0,0	13,0	50,7	36,2
3. The analysis of the current research context in the thesis is					
Original	0,0	11,8	11,8	25,0	51,5
Accurate	0,0	0,0	11,8	36,8	51,5
Useful	0,0	0,0	23,2	11,6	65,2
Consistent	0,0	0,0	0,0	26,9	73,1
Efficient	0,0	0,0	25,0	36,8	38,2
4. How relevant are each of these types of adaptation in eLearning?					
Content	0,0	0,0	13,2	23,5	63,2
Learning flow	0,0	0,0	0,0	50,0	50,0
Interface	0,0	0,0	24,6	39,1	36,2

Feedback	0,0	0,0	13,0	0,0	87,0
Modifications in run-time	0,0	0,0	0,0	63,4	36,6
User grouping	0,0	0,0	12,1	24,2	63,6
Problem solving	0,0	0,0	0,0	41,7	58,3
Information filtering	0,0	0,0	24,6	23,2	52,2
5. How relevant is the solution provided in this thesis and focused on ADAPTIVE LEARNING, related to these eight types of adaptation?					
Content	0,0	0,0	41,0	59,0	0,0
Learning flow	0,0	0,0	0,0	31,0	69,0
Interface	0,0	0,0	42,4	15,3	42,4
Feedback	0,0	0,0	31,4	0,0	68,6
Modifications in run-time	0,0	0,0	14,8	27,9	57,4
User grouping	0,0	0,0	13,8	44,8	41,4
Problem solving	0,0	0,0	50,0	17,3	32,7
Information filtering	0,0	0,0	43,3	28,3	28,3
6. How important are each of these ways of interoperability in eLearning?					
Exportation	0,0	0,0	24,3	11,4	64,3
Importation	0,0	0,0	25,4	11,3	63,4
Embedded play	0,0	0,0	28,8	42,4	28,8
Linked play	15,0	0,0	13,3	43,3	28,3
Sharing of properties and states	0,0	0,0	13,4	0,0	86,6
Integration of information packages	0,0	0,0	12,3	0,0	87,7
7. How relevant is the solution provided in this thesis and focused on INTEGRATION of Units of Learning, related to these ways?					
Exportation	0,0	0,0	0,0	28,6	71,4
Importation	0,0	0,0	0,0	30,5	69,5
Embedded play	0,0	0,0	44,1	27,1	28,8
Linked play	0,0	0,0	43,9	14,0	42,1
Sharing of properties and states	0,0	0,0	12,1	12,1	75,8
8. The solutions (extensions and modifications) presented in this thesis focused on ADAPTIVE LEARNING are					
8.1 For learning designers					
Clear	13,2	0,0	0,0	36,8	50,0
Expressive	0,0	0,0	0,0	50,0	50,0
Useful	0,0	0,0	0,0	29,8	70,2
Re-usable	0,0	0,0	15,0	13,3	71,7
Original	0,0	0,0	0,0	55,9	44,1
Consistent	0,0	0,0	0,0	14,0	86,0
Needed	0,0	0,0	0,0	15,3	84,7
Accurate	0,0	0,0	12,9	43,5	43,5
Implement-able	0,0	0,0	29,3	43,1	27,6
Efficient	0,0	0,0	0,0	70,2	29,8
Flexible	0,0	0,0	29,3	43,1	27,6
8.2 For teachers					
Clear	0,0	0,0	62,9	0,0	37,1
Expressive	0,0	0,0	41,0	29,5	29,5
Useful	0,0	0,0	13,2	38,2	48,5
Re-usable	0,0	12,7	0,0	49,3	38,0
Original	0,0	0,0	23,9	38,8	37,3
Consistent	0,0	0,0	12,3	36,9	50,8
Needed	0,0	0,0	13,6	24,2	62,1
Accurate	0,0	0,0	40,3	59,7	0,0
Implement-able	0,0	0,0	50,0	50,0	0,0
Efficient	0,0	0,0	25,4	74,6	0,0
Flexible	0,0	11,6	24,6	52,2	11,6
8.3 For developers					
Clear	0,0	0,0	0,0	41,5	58,5
Expressive	0,0	0,0	0,0	21,4	78,6
Useful	0,0	0,0	0,0	40,9	59,1
Re-usable	0,0	0,0	20,5	18,2	61,4
Original	0,0	0,0	0,0	37,2	62,8
Consistent	0,0	0,0	0,0	19,5	80,5
Needed	0,0	0,0	0,0	40,0	60,0

	Accurate	0,0	0,0	19,0	61,9	19,0
	Implement-able	0,0	0,0	19,0	19,0	61,9
	Efficient	0,0	0,0	0,0	61,9	38,1
	Flexible	0,0	0,0	20,5	40,9	38,6
8.4 For students						
	Clear	0,0	0,0	67,3	15,4	17,3
	Expressive	0,0	0,0	49,0	33,3	17,6
	Useful	0,0	0,0	17,3	17,3	65,4
	Re-usable	0,0	17,3	17,3	48,1	17,3
	Original	0,0	0,0	35,3	31,4	33,3
	Consistent	0,0	0,0	17,6	64,7	17,6
	Needed	0,0	0,0	15,7	0,0	84,3
	Accurate	0,0	0,0	49,0	34,7	16,3
	Implement-able	0,0	0,0	47,1	52,9	0,0
	Efficient	0,0	0,0	18,4	81,6	0,0
	Flexible	0,0	16,7	0,0	66,7	16,7
9. The solutions (extensions and modifications) presented in this thesis focused on INTEGRATION of Units of Learning are						
9.1 For learning designers						
	Clear	0,0	0,0	15,5	29,3	55,2
	Expressive	0,0	0,0	15,7	33,3	51,0
	Useful	0,0	0,0	14,8	13,1	72,1
	Re-usable	0,0	0,0	15,0	15,0	70,0
	Original	0,0	0,0	0,0	43,1	56,9
	Consistent	0,0	0,0	15,8	0,0	84,2
	Needed	0,0	0,0	0,0	15,0	85,0
	Accurate	0,0	0,0	14,0	43,9	42,1
	Implement-able	0,0	0,0	13,8	56,9	29,3
	Efficient	0,0	0,0	0,0	59,3	40,7
	Flexible	0,0	0,0	0,0	70,5	29,5
9.2 For teachers						
	Clear	0,0	13,4	47,8	0,0	38,8
	Expressive	0,0	13,1	44,3	27,9	14,8
	Useful	0,0	13,4	0,0	37,3	49,3
	Re-usable	0,0	11,9	13,4	37,3	37,3
	Original	0,0	0,0	25,4	37,3	37,3
	Consistent	0,0	13,6	0,0	36,4	50,0
	Needed	0,0	0,0	11,6	0,0	88,4
	Accurate	0,0	0,0	25,0	48,5	26,5
	Implement-able	0,0	0,0	37,9	62,1	0,0
	Efficient	0,0	0,0	11,4	75,7	12,9
	Flexible	0,0	0,0	24,3	62,9	12,9
9.3 For developers						
	Clear	0,0	0,0	21,4	19,0	59,5
	Expressive	0,0	0,0	20,9	20,9	58,1
	Useful	0,0	0,0	19,0	0,0	81,0
	Re-usable	0,0	0,0	21,4	0,0	78,6
	Original	0,0	0,0	0,0	38,1	61,9
	Consistent	0,0	0,0	18,2	20,5	61,4
	Needed	0,0	0,0	0,0	41,9	58,1
	Accurate	0,0	0,0	18,6	41,9	39,5
	Implement-able	0,0	0,0	19,0	40,5	40,5
	Efficient	0,0	0,0	0,0	59,1	40,9
	Flexible	0,0	0,0	0,0	61,0	39,0
9.4 For students						
	Clear	0,0	17,6	49,0	0,0	33,3
	Expressive	0,0	17,6	51,0	0,0	31,4
	Useful	0,0	15,4	0,0	17,3	67,3
	Re-usable	0,0	16,0	18,0	32,0	34,0
	Original	0,0	0,0	34,0	16,0	50,0
	Consistent	0,0	16,3	0,0	51,0	32,7
	Needed	0,0	0,0	16,3	16,3	67,3
	Accurate	0,0	0,0	50,0	32,7	17,3

	Implement-able	0,0	0,0	49,0	51,0	0,0
	Efficient	0,0	0,0	33,3	66,7	0,0
	Flexible	0,0	0,0	33,3	49,0	17,6
10. The evaluation process is						
	Clear	0,0	11,8	0,0	36,8	51,5
	Schedule	0,0	0,0	15,3	44,1	40,7
	Useful	0,0	0,0	11,8	38,2	50,0
	Original	0,0	0,0	51,4	0,0	48,6
	Efficient	0,0	0,0	12,9	58,1	29,0
	Do-able	0,0	0,0	0,0	57,6	42,4
11. Does the solution and-or approach of this thesis show any important drawback?						
	Yes	1,6				
	No	98,4				
12. Does the solution and-or approach of this thesis show any relevant contribution?						
	Yes	100,0				
	No	0,0				
13. Would you like to use these types of solution in a system-specification-authoring tool?						
	Yes	100,0				
	No	0,0				
14. In the future, would you participate in a funded project supporting this contribution?						
	Yes	90,6				
	No	9,4				

Many thanks for your active collaboration. Please, send this filled-in form by 17Dec07 to daniel.burgos@atosresearch.eu