

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA
Departamento de Ingeniería del Software e Inteligencia Artificial



**APLICACIÓN DE TÉCNICAS DE INGENIERÍA DE
LENGUAJES AL CAMPO DEL MODELADO
EDUCATIVO.**

MEMORIA PARA OPTAR AL GRADO DE DOCTOR
PRESENTADA POR

Iván Martínez Ortiz

Bajo la dirección de los doctores

Baltasar Fernández Manjón
José Luis Sierra Rodríguez

Madrid, 2011

ISBN: 978-84-694-5113-7

© Iván Martínez Ortiz, 2011

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA

Departamento de Ingeniería del Software e Inteligencia Artificial



**APLICACIÓN DE TÉCNICAS DE INGENIERÍA DE LENGUAJES AL
CAMPO DEL MODELADO EDUCATIVO**

TESIS DOCTORAL

Presentado por:

Iván Martínez Ortiz

Bajo la dirección de los Doctores:

Baltasar Fernández Manjón

José Luis Sierra Rodríguez

Madrid, 2011

**APLICACIÓN DE TÉCNICAS DE INGENIERÍA DE LENGUAJES AL
CAMPO DEL MODELADO EDUCATIVO**

Memoria que presenta para optar el título de Doctor en Informática

Iván Martínez Ortiz

Bajo la dirección de los Doctores:

Baltasar Fernández Manjón

José Luis Sierra Rodríguez

Universidad Complutense de Madrid

Facultad de Informática

Departamento de Ingeniería del Software e Inteligencia Artificial

Madrid, 2011

A mi padre

Agradecimientos

Al Dr. Baltasar Fernández Manjón, por su apoyo personal y profesional a lo largo de estos años. Su dirección, consejo, aportaciones y paciencia han permitido que esta Tesis saliera adelante.

Así mismo, al Dr. José Luis Sierra Rodríguez, como codirector de esta Tesis, quiero agradecerle su dedicación y disponibilidad plenas para la discusión de esta Tesis, así como su consejo acerca de los peligros de “la grieta” y por rescatarme al caer por la profunda sima.

Por otro lado, agradecer a los también integrantes del grupo <e-UCM>, Dr. Pablo Moreno Ger y Dra. Pilar Sancho Thomas que han aportado diferentes puntos de vista a esta Tesis, permitiendo contrastar las ideas y propuestas de la misma y que siempre han tenido un momento para discutir y analizar alguna de mis ideas. Tengo que agradecer también a los “habitantes” del Aula 16, Javier, Eugenio, Ángel, las entretenidas discusiones científicas y técnicas a las que han aportado su toque especial y que me han permitido llevar el trabajo diario de manera más alegre.

También es importante agradecer el apoyo y los ánimos infundidos por el Dr. Alfredo Fernández Valmayor, antiguo co-director del grupo de investigación <e-UCM>, que actualmente disfruta con su nueva etapa a lo Indiana Jones. Un agradecimiento especial a Carmen Fernández Chamizo por su siempre afable consejo como directora que fue del grupo de investigación ISIA y del cual surgió el propio grupo <e-UCM>.

Quiero agradecer a mi familia el apoyo que me han dado durante mi etapa como estudiante y ahora en mi carrera académica ya que siempre han promovido mi interés investigador. También quiero agradecer a mis amigos Alfonso, Carol, Javi, Naida y Nuria su apoyo a lo largo de este proceso.

Finalmente, pero no por ello menos importante, a Marta por su constante apoyo, implicación y comprensión durante el desarrollo de esta Tesis y en mi carrera académica, y sobre todo por compartir y ser mi compañera en la carrera de la vida.

Diciembre, 2010

Acerca de este documento

Este trabajo es presentado como una recopilación de publicaciones editadas, de acuerdo a la sección 4.4 de la Normativa de desarrollo del Real Decreto 1393/2007, de 29 de Octubre, por el que se establece la ordenación de las enseñanzas universitarias oficiales de la Universidad Complutense de Madrid (Aprobada por el Consejo de Gobierno a 14 de Octubre de 2008, modificado por la Comisión Permanente del Consejo de Gobierno con fecha de 29 de Octubre de 2010, publicado en el BOUC el 20 de Noviembre de 2008).

Los artículos presentados son los siguientes:

- Martínez-Ortiz I., Moreno-Ger P., Sierra J.L., Fernández-Manjón B. Educational Modeling Languages. A Conceptual Introduction and a High-Level Classification. En *Computers and Education E-Learning, From Theory to Practice*. Dordrecht, Países Bajos: Springer; 2007. p. 27-40. ISBN 978-1-4020-4913-2 (libro impreso) ISBN 978-1-4020-4914-9 (libro digital).
- Martínez-Ortiz I., Moreno-Ger P., Sierra J.L., Fernández-Manjón B. Supporting Authoring and Operationalization of Educational Modelling Languages. *Journal of Universal Computer Science*; 28 de Julio 2007; 13(7): 938-947. ISSN 0948-695x (Índice de impacto JCR en 2007: 0,315)
- Martínez-Ortiz I, Sierra JL, Fernández-Manjón B. Enhancing Reusability of IMS-LD Units of Learning: The e-LD Approach. *8th IEEE International Conference on Advanced Learning Technologies (ICALT 2008)*. 1-5 de Julio de 2008; Santander, España. Washington DC, USA: IEEE Computer Society; 2008. pp 402-404.
- Martínez-Ortiz, I., Sierra, J.L., Fernández-Manjón, B., Fernández-Valmayor, A., Language engineering techniques for the development of e-learning applications, *Journal of Network and Computer Applications*, Septiembre 2009, 32(5): 1092-1105. ISSN 1084-8045, DOI: 10.1016/j.jnca.2009.02.005. (Índice de impacto JCR en 2009: 1,111)
- Martínez-Ortiz I., Sierra J.L., Fernández-Manjón B. Authoring and Reengineering of IMS Learning Design Units of Learning. *IEEE Transactions on Learning Technologies* 2(3) , Marzo

2009: 189-202. IEEE computer Society Digital Library. IEEE Computer Society, DOI: 10.1109/TLT.2009.14.

- Martínez-Ortiz I., Moreno-Ger P., Sierra J.L., Fernández-Manjón B. Using DocBook and XML Technologies to Create Adaptive Learning Content. *International Journal of Computer Science and Applications*; Junio 2006; 3(2): 91-108.
- Martínez-Ortiz I., Sierra J.L., Fernández-Manjón B. *Enhancing IMS-LD Units of Learning Comprehension*. 4th International Conference on Internet and Web Applications and Services (ICIW 2009). 24-28 de Mayo de 2009; Venecia, Italia. pp 561-566.
- Martínez-Ortiz I., Sierra J.L., Fernández-Manjón B. *Translating e-learning Flow-Oriented Activity Sequencing Descriptions into Rule-based Designs*. 6th International Conference on Information Technology: New Generations (ITNG 2009). 27-29 de Abril de 2009; Las Vegas, EEUU. Washington DC, EEUU: IEEE Computer Society; 2009. pp. 1108-1113.

De acuerdo a la normativa, este documento también incluye una introducción y un estudio del estado del arte en el dominio. También hay una descripción de los objetivos propuestos para este trabajo y una discusión integrando los contenidos de los ocho artículos y relacionándolos con los objetivos mencionados. Adicionalmente, se presenta una sección con el objetivo de analizar los resultados, así como de resumir unas conclusiones y trabajo futuro. Finalmente, se incluye una bibliografía que integra y complementa todas las referencias de los artículos incluidos.

Resumen

Los Lenguajes de Modelado Educativo (EMLs, del término en inglés *Educational Modeling Language*) permiten a los profesores la formalización de los procesos de enseñanza mediante la creación de un diseño educativo. Estos diseños educativos contemplan tanto los propios contenidos educativos como las actividades complementarias necesarias para lograr un uso eficaz de tales contenidos, así como para mejorar el aprendizaje. Los diseños educativos formalizados mediante un EML tienen la ventaja de poder ser procesados automáticamente mediante herramientas informáticas. El principal tipo de procesamiento de los diseños educativos consiste en la configuración automática del entorno virtual de enseñanza con el fin de dar soporte al proceso de enseñanza representado mediante el EML. Esta representación explícita tiene distintas ventajas, entre las que destaca que los diseños educativos pueden ser reutilizados por otros profesores, bien como ejemplo de buenas prácticas en la enseñanza o, bien como punto de partida para adaptar el diseño educativo a las necesidades específicas de sus alumnos.

Pese al potencial de los EMLs, su uso por parte de los profesores todavía es limitado debido en gran parte, por un lado, a la falta de herramientas sencillas que faciliten su aplicación y, por otro, a la dificultad de su uso como consecuencia de la alta expresividad de estos lenguajes.

El objetivo de esta Tesis consiste en facilitar el uso de los EMLs por parte de los profesores a través de dos líneas principales:

- Acercar la terminología de los EMLs a la terminología utilizada por parte de los profesores.
- Promover y simplificar la reutilización de diseños educativos existentes, poniendo al mismo nivel la reingeniería de diseños educativos y la creación, partiendo de cero, de nuevos diseños.

Esta Tesis propone la creación de EMLs específicos para el proceso de autoría con idea de utilizar una terminología cercana a los profesores y, al mismo tiempo, mantener la compatibilidad con los EMLs estandarizados existentes en la comunidad educativa. Así mismo, propone un modelo de proceso completo en el que los profesores participan activamente en la concepción, diseño y operacionalización de estos EMLs de autoría. Finalmente, esta Tesis aborda la problemática de la reutilización de diseños educativos preexistentes tanto desde el punto de vista metodológico, integrando la reutilización en el proceso de autoría, como desde el punto de vista técnico, proporcionando herramientas de análisis que faciliten la comprensión de los diseños educativos preexistentes.

Este trabajo de Tesis se encuadra dentro del campo de la enseñanza apoyada por la tecnología, que globalmente se denomina por el término en inglés *e-learning*, con la peculiaridad de que aborda los objetivos previamente descritos desde el punto de vista y con herramientas de la

Ingeniería de Lenguajes Software. Con este propósito se consideran los EMLs como Lenguajes Específicos para el dominio educativo. De esta forma, la aproximación de esta Tesis aprovecha un campo, el de la Ingeniería de Lenguajes Software, sólidamente cimentado y con una fuerte tradición, en el que existen métodos sistemáticos, técnicas y herramientas bien definidas que pueden aplicarse al dominio del e-learning.

Estructura del Trabajo

El núcleo de este trabajo es una recopilación de publicaciones editadas, además de un bloque previo que integra y comenta las contribuciones de cada uno de los artículos.

Por tanto, la estructura de este trabajo es como sigue:

- Capítulo 1. Introducción y motivación.
- Capítulo 2. Estudio del dominio.
- Capítulo 3. Objetivos y planteamiento del trabajo.
- Capítulo 4. Discusión y contribuciones.
- Capítulo 5. Conclusiones y trabajo futuro.
- Capítulo 6. Recopilación de todas las publicaciones editadas.

Índice general

Agradecimientos	VII
Acerca de este documento.....	IX
Resumen.....	XI
Estructura del Trabajo.....	XIII
Índice general.....	XV
Capítulo 1 Introducción.....	19
1.1. Motivación de la Investigación.....	19
1.2. Objetivos y Planteamiento de la Línea de Investigación.....	20
Capítulo 2 Estudio del dominio.....	23
2.1. Lenguajes de Modelado Educativo	23
2.1.1. El concepto de Modelado Educativo.....	23
2.1.2. Los Lenguajes de Modelado Educativo	25
2.1.3. Clasificación de los principales EMLs.....	28
2.1.4. IMS Learning Design.....	37
2.1.5. Herramientas de soporte a IMS-LD	41
2.1.6. Motores de ejecución compatibles con IMS-LD.....	45
2.1.7. Reproductores de IMS-LD.....	46
2.1.8. Otras iniciativas y proyectos de investigación relacionados con IMS-LD.....	48
2.2. Lenguajes de Procesos de Negocio	50
2.2.1. Lenguajes de Modelado Educativo y Gestión de Procesos de Negocio.....	50
2.2.2. Lenguajes de Descripción de Procesos de Negocio	50
2.3. Ingeniería de Lenguajes Software.....	53
2.3.1. Lenguajes de Modelado Educativo e Ingeniería de Lenguajes Software.....	53
2.3.2. Lenguajes Específicos de Dominio.....	54
2.3.3. Desarrollo de Software Dirigido por Modelos	57
2.4. A modo de conclusión.....	59
Capítulo 3 Objetivos y Planteamiento del Trabajo.....	63
3.1. Objetivos de la Tesis.....	63
3.1.1. Facilitar la adopción de los lenguajes de modelado educativo.....	64

3.1.2.	Fomentar la reutilización de diseños educativos.....	67
3.2.	Planteamiento del Trabajo: La Propuesta <e-LD>.....	68
3.2.1.	Facilitar la adopción de los lenguajes de modelado educativo.....	69
3.2.2.	Fomentar la reutilización de diseños educativos.....	71
3.3.	A modo de conclusión.....	73
Capítulo 4	Discusión de las contribuciones de los artículos	75
4.1.	Lenguajes de Modelado Educativo como Lenguajes Específicos de Dominio.....	76
4.1.1.	Análisis del Dominio de los Lenguajes de Modelado Educativo.....	76
4.1.2.	Gestión de Contenidos Mediante Lenguajes de Marcado Descriptivos Específicos de Dominio.....	77
4.2.	Ingeniería de Lenguajes Software y Lenguajes de Modelado Educativo.....	78
4.3.	El Enfoque <e-LD>.....	79
4.4.	Autoría y Reutilización de Unidades de Aprendizaje en <e-LD>.....	80
Capítulo 5	Conclusiones y Trabajo Futuro.....	83
5.1.	Principales Aportaciones.....	83
5.1.1.	Contextualización de los Lenguajes de Modelado Educativo en el marco general de los Lenguajes Específicos de Dominio.....	84
5.1.2.	Aplicación de la Ingeniería de los Lenguajes Software al desarrollo sistemático de herramientas de soporte de Lenguajes de Modelado Educativo.....	84
5.1.3.	Arquitectura Flexible para sistemas e-learning dirigidos por Lenguajes de Modelado Educativo	85
5.1.4.	Herramienta <e-LD> Author para la autoría y la reutilización de diseños educativos en IMS-LD	86
5.2.	Trabajo Futuro	87
5.2.1.	Completar <e-LD> Author y mejorar su usabilidad	88
5.2.2.	Inclusión del nivel C de IMS-LD en <e-LD> Author	88
5.2.3.	Aplicación de la aproximación <e-LD> a otros lenguajes de intercambio.	88
5.2.4.	Integración de <e-LD> Author con motores de ejecución de IMS-LD.....	89
5.2.5.	Mejora del enfoque de importación y reingeniería en <e-LD>.....	89
5.2.6.	Creación de un <i>DSL Toolkit</i> para EMLs.	89
5.2.7.	Creación de un motor de ejecución IMS-LD basado en tecnologías de gestión de procesos de negocio.	90
5.2.8.	Aplicación al Dominio de los Juegos y las Simulaciones Educativas.....	90
Capítulo 6	Artículos Presentados	91

6.1.	Educational Modeling Languages: A Conceptual Introduction and a High-Level Classification.....	92
6.2.	Supporting the Authoring and Operationalization of Educational Modelling Languages	108
6.3.	Enhancing Reusability of IMS-LD Units of Learning: The e-LD Approach.....	119
6.4.	Language Engineering Techniques for the Development of E-Learning Applications	123
6.5.	Authoring and Reengineering of IMS Learning Design Units of Learning	144
6.6.	Using DocBook and XML Technologies to Create Adaptive Learning Content in Technical Domains	159
6.7.	Enhancing IMS-LD Units of Learning Comprehension	177
6.8.	Translating e-learning Flow-Oriented Activity Sequencing Descriptions into Rule-based Designs.....	184
	Bibliografía	191

Capítulo 1

Introducción

1.1. Motivación de la Investigación

En la enseñanza apoyada por la tecnología, que globalmente se denomina por el término en inglés *e-learning*, se ha producido una gran revolución con la nueva forma de crear contenidos que supone los objetos de aprendizaje (LO, del término en inglés *learning objects*). La principal ventaja de esta nueva forma de creación de contenidos es que permite crear cursos mediante combinación de contenidos previamente existentes; es decir potencia la reusabilidad y la interoperabilidad (Fernández-Manjón et. al., 2007). No obstante, y a pesar de las ventajas que aportan los LO en e-learning, existe también un amplio consenso entre los educadores acerca de que la creación y presentación de materiales educativos de gran calidad no es suficiente para obtener una experiencia educativa plena y satisfactoria: es igualmente importante la planificación de las otras actividades (tutorías, exámenes, lectura de libros, etc.) que el estudiante debe llevar a cabo para utilizar los contenidos y conseguir los objetivos educativos propuestos por el profesor.

De este análisis surge el concepto de Lenguaje de Modelado Educativo (EML, del término en inglés *Educational Modeling Language*) como nueva piedra angular del e-learning, ya que se pretende que los EMLs puedan ser utilizados por los profesores para formalizar los procesos de enseñanza (Rawlings et al., 2002) (Koper, 2000). De esta manera, las descripciones resultantes de este proceso de formalización reflejan no sólo los aspectos de contenidos (que siguen estando en los LO) sino también las actividades de la formación. Así mismo, al estar dichas descripciones representadas en un lenguaje formal, pueden ser interpretadas y ejecutadas por las computadoras.

No obstante, es necesario destacar que estos lenguajes de modelado educativo, a pesar de su potencial, se encuentran todavía más en los estadios de investigación y prueba académica que en los estadios de aplicación directa e inmediata a gran escala en el día a día de los sistemas y escenarios e-learning. Hay razones de diversa índole que pueden explicar esta situación y que van desde educadores que siguen teniendo dudas sobre su aplicabilidad práctica hasta la falta de herramientas suficientemente maduras y sencillas para profesores que no tengan un adecuado

conocimiento técnico. A pesar de esto, un sentir general en la comunidad e-learning es que cualquier avance que se produzca en este campo será de máxima relevancia y que, por tanto, a pesar de sus limitaciones actuales, es necesario continuar invirtiendo esfuerzo en mejorar la aplicabilidad práctica de los EMLs (Griffiths et. al., 2005). En esta Tesis se aborda dicha línea de investigación desde el punto de vista más técnico y de aplicación, complementado, de esta manera, otros trabajos más centrados en las teorías educativas subyacentes (Mayes & de Freitas, 2005).

1.2. Objetivos y Planteamiento de la Línea de Investigación

Como se ha sugerido anteriormente, las grandes expectativas creadas por la aparición de los EMLs para lograr sistemas de e-learning mucho más versátiles y funcionales no se han visto completamente cumplidas. Actualmente existen todavía distintos obstáculos que impiden adoptar de manera generalizada los EMLs y, por tanto, obtener los beneficios asociados con los mismos: sistemas más centrados en los alumnos, en las actividades educativas y en modelos educativos que puedan ser reutilizables e interoperables entre distintos sistemas de e-learning (Koper, 2001). Algunos de estos problemas como, por ejemplo, la excesiva complejidad de uso de los EMLs por parte de los instructores (debido a la alta expresividad), son problemas que consideramos hay que abordar desde el punto de vista metodológico (Burgos et al., 2007). Otros problemas como, por ejemplo, la falta de herramientas de autoría para los EMLs o la falta de reproductores en los sistemas e-learning que soporten las propuestas estándar de EMLs, son inconvenientes que deben abordarse desde el punto de vista tecnológico (Griffiths et. al., 2005).

Los objetivos principales de esta línea de investigación son:

- El primer objetivo es acercar los EMLs a los instructores (Martínez-Ortiz et al., 2008b). Puesto que los instructores son los usuarios finales de estos lenguajes, para facilitar su uso y promover su adopción es necesario que los EMLs se ajusten a la terminología empleada por dichos instructores y a la experiencia previa de los mismos. Al mismo tiempo, se debe mantener la compatibilidad con las propuestas estandarizadas de EMLs para reutilizar la experiencia, herramientas e integración con plataformas existentes compatibles con las propuestas estándar (Martínez-Ortiz et al., 2009).
- El segundo objetivo es promover la reutilización de diseños educativos existentes. La reutilización de diseños educativos previamente creados no suele ser directa, ya que es necesario comprender el diseño y realizar un proceso de adaptación teniendo en cuenta las necesidades de los estudiantes a los que está dirigido el diseño. Por tanto, el proceso de reutilización debe recibir una atención equiparable al proceso de autoría de diseños educativos (Martínez-Ortiz et al., 2009c).

Como caso de estudio para poner en práctica los avances desarrollados en esta línea de investigación se ha seleccionado IMS Learning Design (IMS-LD) (IMS, 2003) (Koper & Tattersall, 2005) como EML representativo. La razón para seleccionar IMS-LD como caso de estudio se debe a que este EML goza de un alto grado de estandarización, así como de una alta capacidad expresiva. No obstante, continúan existiendo serias dificultades para su aceptación por parte de la comunidad de instructores, así como para su inclusión generalizada en sistemas e-learning de uso habitual.

El primer objetivo de esta línea de investigación de acercar los EMLs a los instructores se aborda desde tres frentes:

- Integración de lenguajes de modelado educativo de autoría y lenguajes de modelado educativo de intercambio (Martínez-Ortiz et al., 2008b). Por un lado, los EMLs de autoría son lenguajes específicos de dominio más cercanos a la notación y vocabulario que utilizan los instructores para describir sus diseños educativos. Estos EMLs de autoría se concentran en los conceptos clave desde el punto de vista educativo, obviando la mayor parte de los detalles técnicos necesarios para su procesamiento automático. Además, estos EMLs de autoría tienen una expresividad limitada y adaptada a las necesidades específicas de un instructor o de una comunidad concreta. Por otro lado, los EMLs de intercambio se corresponden con propuestas estándar de EMLs que tienen una gran expresividad para dar soporte a una gran cantidad de escenarios educativos, y, por tanto, una mayor independencia del dominio. Además, en los EMLs de intercambio es difícil abstraerse de algunas de las características tecnológicas de más bajo nivel, que son necesarias para simplificar la interoperabilidad entre distintas herramientas y plataformas.
- Propuesta de un modelo de proceso completo que integre a instructores y desarrolladores. Este enfoque integrador que combina los dos tipos de EMLs, debe incluir un modelo de proceso completo que, por un lado, contemple las distintas fases de provisión, uso y evolución del EML de autoría y que, por otro lado, determine el papel a realizar por los principales actores (los instructores y los desarrolladores).
- Propuesta de una notación gráfica para el EML de autoría. Las notaciones gráficas se utilizan con éxito en distintos dominios y, en muchos casos, con varios propósitos. Por ejemplo, una notación gráfica puede servir, por un lado, como herramienta de diseño, y, por otro lado, como herramienta de documentación o análisis. En base a experiencias previas, como por ejemplo el sistema LAMS (Dalziel, 2006) desarrollado en la Macquarie University de Australia, existe evidencia de que el uso de la notación gráfica facilita la creación de nuevos diseños educativos complejos y reduce la carga cognitiva del instructor. Además, los diagramas creados con la notación gráfica son de mayor utilidad durante el proceso de reutilización de un diseño

educativo (Hernández-Leo et. al., 2006). Con la notación gráfica, el instructor que intenta reutilizar el diseño (es decir, un instructor que no ha sido el creador del mismo) obtiene una idea general de la estructura de dicho diseño, simplificando, por tanto, el análisis y la comprensión del mismo.

Finalmente, el segundo objetivo de promover la reutilización de diseños educativos existentes se aborda desde los siguientes frentes:

- Enfoque completo del proceso de autoría y reutilización de diseños educativos. Se plantea un enfoque completo del proceso de autoría que incluye los distintos aspectos: (i) la propuesta de una notación gráfica de alto nivel, (ii) el planteamiento de una metodología de diseño educativo orientado al flujo de actividades, (iii) la compatibilidad con estándares de e-learning mediante exportación automática y, (iv) el proceso semiautomático de importación desde representaciones XML de diseños educativos IMS-LD preexistentes. En este último aspecto del proceso de importación semiautomática se identifica como necesaria la colaboración de un experto en la tecnología (tanto en XML como en el estándar IMS-LD) que ayude a representar en la notación aquellos conceptos más complejos que actualmente no pueden ser importados de forma automática.
- Herramientas de análisis de diseños educativos para facilitar su reutilización. Las herramientas de autoría están específicamente diseñadas para simplificar el proceso de autoría y, habitualmente, facilitan también la reutilización de diseños previos realizados con la misma herramienta y notación. Sin embargo, el soporte para simplificar la reutilización de un diseño educativo preexistente y representado en otra notación es un proceso mucho más difícil. El objetivo de este frente es proporcionar herramientas que ayuden a dicha comprensión en alguno de los aspectos que se han identificado como más complejos de entender, por ejemplo, el análisis de las complejas relaciones que se establecen entre los elementos que intervienen en la representación del diseño.

Para lograr estos objetivos, en la investigación llevada a cabo en esta Tesis se adoptarán métodos, técnicas y herramientas de la Ingeniería de Lenguajes Software como elementos básicos en los que fundamentar las propuestas realizadas.

Capítulo 2

Estudio del dominio

Como se ha indicado en el capítulo anterior, esta Tesis aborda la problemática relativa a facilitar y simplificar tanto el uso de los EMLs por parte de los instructores durante la producción de diseños educativos, como la reutilización de diseños educativos previamente existentes. Para ello se propone utilizar técnicas de Ingeniería de Lenguajes Software como instrumento básico. De esta forma, la investigación realizada se apoya, por una parte, en las propuestas relativas a los EMLs existentes en la actualidad (Rawlings et al., 2002). Por otra parte, también se basa en los aspectos relativos a la Ingeniería de Lenguajes Software y, más concretamente, a los lenguajes específicos de dominio y a las tendencias modernas en el desarrollo de dichos lenguajes mediante enfoques dirigidos por modelos (Kleppe, 2008). Por último, en esta investigación se han tenido en cuenta los trabajos realizados en el dominio de los lenguajes de modelado de procesos de negocio, debido a las importantes similitudes entre este dominio y el dominio de los EMLs (Aalst & Kees, 2004).

De esta forma, este capítulo presenta un estudio y análisis de las tecnologías y enfoques que se han considerado más relevantes para encuadrar adecuadamente este trabajo. Para ello, se comienza revisando los aspectos más relevantes del dominio de los EMLs. Seguidamente se analizan las características más importantes de los lenguajes de modelado de procesos de negocio. Por último, el capítulo revisa los fundamentos y características más relevantes de la Ingeniería de Lenguajes Software, incidiendo, para ello, en el diseño de lenguajes específicos de dominio y en las técnicas dirigidas por modelos.

2.1. Lenguajes de Modelado Educativo

2.1.1. El concepto de Modelado Educativo

El concepto de modelado educativo es muy amplio y es previo a su uso en el campo del e-learning. Ya sea en enseñanza presencial, como en enseñanza a distancia, ha habido muchos

esfuerzos para planificar y documentar el proceso utilizado para enseñar a los alumnos (Dick et al., 2000). No obstante, en la mayoría de los casos, se han realizado descripciones informales (a veces descritas como “recetas”) o estructuradas mediante fichas (patrones o plantillas), pero no se han observado iniciativas exitosas, ampliamente aceptadas, de formalización y documentación rigurosa y estándar del proceso educativo. En ese sentido se pueden distinguir, al menos, tres categorías de diseños (Martínez-Ortiz et al. 2007):

- Los diseños informales dónde sólo se proporcionan indicaciones sobre el proceso educativo y que, normalmente, contemplan los contenidos, el contexto y las estrategias a utilizar. No obstante, dichas descripciones no siguen un patrón común ni tienen porqué abordar los mismos aspectos.
- Los diseños estructurados en base a plantillas documentadas que tienen una serie de apartados que describen un conjunto fijo de aspectos y que los instructores cumplimentan en todos los casos. De esta forma se obtienen descripciones más regulares, pero normalmente no se ponen limitaciones o normas estrictas acerca de cómo rellenar dichos apartados. En el mejor de los casos, las plantillas se acompañan con una guía metodológica que describe cómo rellenarlas.
- Los diseños formales que se expresan mediante un lenguaje informático formal y específico de dominio. Este lenguaje proporciona una sintaxis que clarifica qué se puede describir y una semántica que determina el significado de los diseños formalizados. Estos diseños formales, aunque mucho más trabajosos de crear, son susceptibles de automatización. Esto significa, por ejemplo, que se puede comprobar que los diseños son correctos y que es factible crear un sistema informático que los ejecute automáticamente.

Desde el punto de vista del alcance del modelado se pueden distinguir los modelados especializados, que pretenden cubrir sólo un ámbito o tipo determinado de actividades educativas (e.g. WebQuest) y los modelados genéricos, que pretenden cubrir cualquier tipo de situación educativa, tanto por el dominio como por el tipo de actividades o medios utilizados.

En e-learning se comienza a hablar de modelado educativo cuando se deja de considerar que los contenidos (y, por tanto, los objetos de aprendizaje) son el centro y elemento principal de un aprendizaje en el que sólo se tiene en cuenta el escenario de un alumno individual accediendo al contenido. De ahí, se ha pasado a una visión más global en la que se tratan de especificar los procesos educativos de una forma más completa, en base a las condiciones en las que éstos se realizan, y también en base a las actividades que tienen que llevar a cabo tanto los alumnos como los profesores para lograr unos determinados objetivos de aprendizaje. La idea es pasar de sistemas basados o centrados en contenidos a sistemas más orientados a actividades y aprendizaje activo (aunque la calidad de los contenidos sigue siendo imprescindible) que permitan incrementar

las posibilidades que ofrecen los entornos de gestión de e-learning. Las ideas subyacentes a este enfoque son (Britain, 2004):

- Las personas aprenden mejor cuando están implicadas activamente en la realización de una actividad (actividad de aprendizaje). El aprendizaje es un proceso activo, que requiere esfuerzo, y en el cual no todos los alumnos tienen la misma capacidad de aprender por sí mismos. Este aprendizaje puede verse facilitado si se proporciona algún tipo de guía o soporte (estrategia pedagógica) que implique o motive a los alumnos (trabajo en grupo, aprendizaje basado en problemas, etc).
- Las actividades de aprendizaje se pueden secuenciar y organizar para lograr un aprendizaje más efectivo. Este secuenciamiento es lo que se ha denominado flujo de aprendizaje. El aprendizaje se mejora no sólo si se tienen actividades que impliquen a los estudiantes, sino también si se diseña de forma cuidadosa su secuenciación en el tiempo o la duración de dichas actividades. De esta forma, se pueden considerar, por ejemplo, distintas “rutas” de aprendizaje, como tareas que puedan ser realizadas en paralelo o trabajos que deben completarse en subgrupos antes de continuar con el desarrollo del curso.
- Los diseños educativos se pueden describir de una forma consistente (formal) y transferible para facilitar que puedan ser compartidas y reutilizadas. Aquí surge el problema de cómo describir una estrategia de enseñanza de un modo lo suficientemente abstracto como para que sea útil en un contexto diferente para el que fue creado, pero que a la vez sea suficientemente detallada como para que se pueda ponerse en práctica sin perder su valor pedagógico. Además, dicha descripción debe ser procesable automáticamente por una computadora.

Parece ampliamente aceptado que el modelado educativo, a pesar de sus dificultades, tiene una serie de ventajas. Por un lado, permite que los profesores formalicen sus diseños educativos de modo que queden reflejadas qué actividades se realizan y cómo se organizan dichas actividades. Por otro lado, cuando un diseño ha probado su eficacia, puede ser compartido con otros docentes o archivado para un uso o consulta posterior.

2.1.2. Los Lenguajes de Modelado Educativo

La generalización del término EML en e-learning proviene del trabajo desarrollado en la Universidad Abierta de los Países Bajos (OUNL) durante finales de los años 90. El grupo de investigación liderado por el Profesor Rob Koper analizó los sistemas de gestión de la enseñanza (LMSs, de su término en inglés *Learning Management Systems*) que existían y que eran los más utilizados en aquella época, intentando identificar los problemas y defectos de dichos sistemas. En particular se identificó como principal problema la falta de aplicación de la teoría instruccional y

del aprendizaje dentro de los mismos. Como resultado se desarrolló y puso en práctica una propuesta basada en la definición de un lenguaje específico de dominio llamado Educational Modeling Language (OUNL-EML) (para evitar ambigüedades se denominará a este lenguaje OUNL-EML a lo largo de este capítulo, en lugar de simplemente EML).

En un estudio realizado por el *CEN/ISSS WS/LT Learning Technology Workshop* acerca de los Lenguajes de Modelado Educativo (Rawlings et al., 2002) se define el concepto de EML como:

Modelo de información semántico y su vinculación, que describen el contenido y el proceso dentro de una “Unidad de Aprendizaje” desde una perspectiva pedagógica y con el objetivo de dar soporte a la reutilización y la interoperabilidad.

De esta definición pueden extraerse los siguientes conceptos principales:

- *Modelo de Información Semántico*. Un modelo de información semántico es un metamodelo (conceptualización) de un dominio de discurso. En este caso se trata de un metamodelo que describe el proceso de enseñanza/aprendizaje.
- *Modelo de información y vinculación*. La “vinculación” de un EML es una formalización lingüística del modelo semántico. Habitualmente, esta formalización se realiza mediante la definición de un lenguaje específico de dominio basado en las tecnologías XML a fin de conseguir una vinculación o representación directamente procesable en el ordenador.
- *Unidad de Aprendizaje*. El concepto de Unidad de Aprendizaje (UoL, del término en inglés *Unit of Learning*) es el punto clave de los EMLs. En palabras del Profesor Koper (Koper, 2001):

Una UoL (también conocida como unidad de estudio) es la menor unidad que proporciona eventos educativos a los estudiantes, satisfaciendo uno o más objetivos educativos interrelacionados.

De esta forma, en los EMLs se pasa del concepto de LO como elemento constructivo básico y atómico a otro de mayor granularidad que es la UoL y que no sólo agrupa contenidos. Por tanto, una UoL no puede dividirse sin perder su propia semántica orientada al logro de los objetivos educativos. Una UoL puede ser un curso, un taller, una práctica, una titulación completa, etc. Cada UoL define el modelo instructivo y el entorno donde se realiza. Este entorno está caracterizado por los recursos materiales (que pueden ser LOs) y los servicios (v.g. foro, chat, videoconferencia, e-mail) que serán utilizados durante la puesta en ejecución de la UoL.

- *Perspectiva Pedagógica.* Un EML debe ser relativamente independiente de las teorías instruccionales, de manera que el profesor o el diseñador instruccional pueda decidir cuál de estas teorías desea aplicar. De esta forma, el EML no trata de limitar la expresividad del docente o de imponer una visión determinada de cómo debe realizarse la enseñanza.
- *Reutilización e Interoperabilidad.* La idea detrás de los EMLs no es sólo permitir a las aplicaciones informáticas interpretar las UoLs creadas mediante dichos lenguajes sino también promover la reutilización, por ejemplo, aquellas UoLs que hayan tenido una aplicación exitosa. Además, también permiten el intercambio de estas unidades de aprendizaje entre distintos sistemas de e-learning sin tener en cuenta cómo el sistema de información implementará finalmente la semántica del modelo definido.

Además de estos conceptos básicos asociados a los EMLs, el Profesor Koper (Koper, 2000) identifica las siguientes características deseables que debería cumplir un EML:

- Un Lenguaje de Modelado Educativo debe estar definido formalmente y tiene que poder ser procesable por un sistema informático, de manera que las UoLs creadas con dicho lenguaje puedan ser interpretados por aplicaciones informáticas.
- Un Lenguaje de Modelado Educativo tiene que ser pedagógicamente neutral. Como ya se ha indicado anteriormente, el lenguaje no debe imponer restricciones a la forma de enseñar y, por tanto, debe permitir la aplicación de las distintas estrategias pedagógicas que el educador considere oportunas en la concepción de las UoLs.
- Un Lenguaje de Modelado Educativo debe permitir a los diseñadores crear UoLs completas que incluyan las actividades a realizar por el estudiante (¿qué hacer?), las personas involucradas en dichas actividades (¿con quién?), y el entorno donde se llevarán a cabo las actividades (¿qué materiales son necesarios?, ¿qué herramientas?, etc.).
- Una UoL creada utilizando un EML debería ser perdurable, es decir, resistente a los cambios y evoluciones tecnológicas, así como a cambios de plataformas, puesto que su propósito es facilitar la reusabilidad e interoperabilidad entre distintos sistemas y herramienta.

A modo de resumen, los EMLs son lenguajes que permiten describir UoLs, las cuales a su vez describen el proceso de aprendizaje como un todo (y no sólo centrado en los contenidos como se hace en el caso de los objetos de aprendizaje).

Además, otra característica añadida de estos lenguajes es que proporcionan un mecanismo para la comunicación dentro de una organización entre el personal técnico de soporte y el personal no técnico (normalmente los educadores) que es necesaria para la adecuada operacionalización del EML. Ahora las UoLs son completas, de modo que el personal técnico puede saber qué es lo que

está planificado y ayudar a resolver las incidencias que pudieran producirse (v.g. no disponibilidad de un recurso).

2.1.3. Clasificación de los principales EMLs

A partir de las diferentes iniciativas desarrolladas en base a los principios de los lenguajes de modelado educativo previamente mencionados, es posible crear una clasificación estos similar la propuesta en (Vantroys, 2003):

- *Lenguajes Específicos*. En esta categoría se encuentran los lenguajes que, aún sin cumplir estrictamente todas las características de un EML, permiten a los diseñadores describir las etapas del proceso de aprendizaje utilizando una metodología específica. En particular, dentro de esta categoría podemos destacar aquellos lenguajes aplicados a la metodología de enseñanza basada en la resolución de problemas, mediante el planteamiento de preguntas, y a la recolección de soluciones y respuestas.
- *Lenguajes de Estructuración de Contenidos*. Esta categoría está formada por aquellos lenguajes que permiten a los diseñadores organizar los recursos educativos en una secuencia, siempre teniendo en cuenta las necesidades del estudiante y la interacción con el propio contenido, a fin de mejorar la experiencia educativa.
- *Lenguajes de Actividad*. En esta categoría se encuentran los lenguajes que principalmente están enfocados a la organización de actividades en general (utilizando computadoras o no) durante el proceso de aprendizaje.

En la tabla Tabla 2.1 se clasifican, de acuerdo a las tres categorías propuestas, los EMLs que aparecen el estudio de Rawlings (Rawlings et al., 2002) y los EMLs más relevantes en la literatura que aparecieron posteriormente a dicho estudio. En los siguientes puntos se darán más detalles acerca de cada uno de estos lenguajes.

Tabla 2.1. Clasificación de alto nivel de los Lenguajes de Modelado Educativo más relevantes.

Tipo	Lenguaje	Sitio web
Lenguajes Específicos	Tutorial Markup Language (TML)	http://www.ilrt.bris.ac.uk/netquest
	<e-Adventure>	http://e-adventure.e-ucm.es
	IMS Question & Test Interoperability (IMS QTI)	http://www.imsglobal.org/question/
Lenguajes de estructuración de contenidos	TARgeted Reuse and GENeration of TEACHing Materials (Targeteam)	http://www.targeteam.net
	Learning Material Mark-up Language (LMML)	No disponible
	ARIADNE Course (Curriculum) Description Format (A-CDF)	http://www.ariadne-eu.org/

	AICC Course Data Model (AICC/CMI)	http://www.aicc.org
	IMS Simple Sequencing (IMS SS)	http://www.imsglobal.org/simplesequencing/
	ADL Sharable Content Object Reference Model 2004 (SCORM)	http://www.adlnet.gov/scorm/
	eLML	http://www.elml.ch/
	<e-DocBook>	http://e-docbook.e-ucm.es
Lenguajes de Actividad	Educational Modeling Language - Open University of the Netherlands (EML-OUNL)	http://celstec.org/content/educational-modelling-language
	PALO	http://sensei.ieec.uned.es/palo/
	Educational Environment Modeling Language (E2ML)	http://www.istituti.usilu.net/botturil/publications.htm
	Méthode d'ingénierie d'un système d'apprentissage (MISA)	http://www.licef.telug.quebec.ca/gp/
	XEDU	No disponible
	<e-LD>	http://www.e-ucm.es/projects/eld/
	FlexoLD	No disponible
	PoEML	http://www-gist.det.uvigo.es/~mcaeiro/thesis/
	LAMS	http://www.lamsfoundation.org
	Learning Design Visual Sequence (LVDS)	http://www.learningdesigns.uow.edu.au/
	Learning Design Language (LDL)	No disponible
	IMS Learning Design (IMS-LD)	http://www.imsglobal.org/learningdesign/

2.1.3.1. Lenguajes Específicos

Tutorial Markup Language (TML) (Brickley, 1995) es una extensión de HTML para crear preguntas. TML está diseñado para separar el contenido de la pregunta de su presentación. El formato de los archivos TML es texto plano, pudiendo ser generados a partir de otros formatos y otras preguntas que se encuentren almacenadas en una base de datos.

IMS Question and Test Interoperability (IMS-QTI) es una propuesta desarrollada por el consorcio internacional IMS (Instructional Management Systems – Global Learning Consortium) para crear bancos de preguntas y de evaluaciones (IMS, 2006b). El principal objetivo de IMS-QTI es permitir el intercambio de evaluaciones y de la información asociada a las evaluaciones entre distintos LMSs. En las evaluaciones creadas con IMS-QTI existe una división clara entre las preguntas en sí mismas (¿qué es lo que se pregunta?), la forma en la que se presentan dichas preguntas al alumno y el modo en el que se evalúan las respuestas dadas por el mismo. IMS-QTI permite crear test interactivos, los cuáles pueden incluir pistas (información para ayudar a los alumnos). También es posible crear plantillas de exámenes que pueden ser instanciadas cuando los estudiantes realizan el examen, creando diferentes exámenes a partir de la misma plantilla.

<e-Adventure> (Moreno-Ger et al., 2007) es un proyecto que propone un modelo de desarrollo de videojuegos educativos que pueden distribuirse directamente a través de un LMS (siempre que el LMS cumpla con los estándares de IMS Content Packaging). Con <e-Adventure> es posible crear aventuras gráficas y simulaciones con estructura de juego que tengan un objetivo educativo. Esta iniciativa está compuesta por dos conceptos principales: el motor <e-Adventure> y el lenguaje <e-Adventure>:

- El motor de <e-Adventure> ejecuta los juegos que están representados (o codificados) mediante el lenguaje <e-Adventure>. Este motor es capaz de comunicarse con un sistema de e-learning o LMS permitiendo que el juego pueda informar al LMS acerca del progreso del alumno al interactuar con el mismo.
- El lenguaje <e-Adventure> es un lenguaje específico de dominio que permite a un profesor, con pocos conocimientos tecnológicos la definición de juegos educativos (Moreno-Ger et al., 2008). Este lenguaje permite la codificación del *storyboard* del juego, así como la definición de los personajes y objetos con los que podrá interactuar el alumno. Así mismo, también es posible definir de manera simple las condiciones de las que dependen las distintas acciones que pueden llevarse a cabo en el juego. Finalmente, el lenguaje también incluye construcciones que permiten la monitorización de características importantes desde el punto de vista pedagógico (v.g. si el alumno se queda bloqueado en alguna parte del juego o si toma decisiones erróneas que implican algún error de concepto) e incluso la creación de juegos que se adaptan a las características específicas del usuario. De esta forma, las características distintivas de <e-Adventure> son que tanto el motor como el lenguaje <e-Adventure> permiten crear juegos adaptativos e incluyen mecanismos que monitorizan e informan acerca de la actividad de los estudiantes dentro del juego.

2.1.3.2. Lenguajes de Estructuración de Contenidos

TARgeted Reuse and GENeration of TEACHing Materials (Targeteam) permite la creación y el mantenimiento (uso y reutilización) de contenidos educativos (Koch, 2002). Este EML permite el uso de los materiales en diferentes situaciones y dominios pedagógicos (primaria, secundaria y nivel universitario). Haciendo uso de Targeteam, es posible crear las notas de clase, además de otros contenidos como aclaraciones, explicaciones y ejemplos. Este lenguaje está enfocado al uso de tecnologías XML, como TeachML, e introduce el concepto de tema (*issue*) como UoL.

Learning Material Markup Language (LMML) está basado en un metamodelo que puede encajar en distintos dominios de aplicación. LMML ha sido diseñado como aplicación del meta-lenguaje XML para la descripción de los contenidos educativos (Weitl et al., 2002). Estos contenidos educativos están estructurados en módulos que, a su vez, pueden estar estructurados en

sub-módulos. LMML se basa en una estructuración del contenido educativo de manera jerárquica y modular, donde los contenidos creados con LMML pueden ser adaptados a diferentes situaciones de aprendizaje y a distintos tipos de estudiantes. Utiliza el concepto de curso (*course*) como UoL.

ARIADNE Course Description Format (A-CDF) es un EML que permite la creación de cursos en línea (Verbert & Duval, 2004). Un curso en A-CDF consiste en documentos XML que serán utilizados conjuntamente con un LMS que será el que finalmente generará los cursos (Durm et al., 2001). A-CDF pone especial interés en el contenido y en su agregación, siendo además lo suficientemente expresivo como para describir el proceso de aprendizaje de acuerdo con un modelo pedagógico. El desarrollo con A-CDF se realiza a través de un conjunto de herramientas construidas en el seno del consorcio ARIADNE (editores curriculares, LMS, KPS). Este lenguaje establece el concepto de curso (*course*) como UoL.

IMS Simple Sequencing (IMS-SS) define mecanismos para poder representar el orden de las actividades dentro de una experiencia educativa, de manera que un LMS pueda secuenciar actividades discretas de forma consistente (IMS, 2003b). Los diseñadores instruccionales o los desarrolladores de contenido declaran el orden relativo en el cuál se presentan al alumno los elementos de contenido, y las condiciones bajo las cuales una pieza de contenido se selecciona, se muestra o se omite durante la presentación. Utiliza el concepto de actividad de aprendizaje (*learning activity*) como UoL.

AICC Course Data Model (AICC/CMI) contiene toda la información necesaria para describir un curso (AICC/CMI_CMI001, 2004). Este formato puede intercambiarse entre distintos LMSs mediante herramientas de importación y exportación, utilizando el concepto de Unidades de Asignación (*Assignable Units*) como unidad de intercambio. La información generada durante la interacción del alumno con las unidades de asignación también es almacenada por el LMS. AICC/CMI utiliza el concepto de curso (*course*) como UoL. El secuenciamiento dentro del curso se controla mediante el uso de los requisitos que deben satisfacer los estudiantes antes de acceder a una nueva unidad de asignación.

ADL Sharable Content Object Reference Model (SCORM) representa un modelo de coordinación que tiene el objetivo de proporcionar una colección de prácticas estandarizadas susceptibles de ser ampliamente aceptadas y ampliamente implementadas en entornos de e-learning (ADL, 2009). De hecho, el modelo SCORM puede ser considerado como un *perfil de aplicación* (del término en inglés *application profile*) de estas prácticas ya que su definición se apoya y concreta algunos aspectos de otras especificaciones más generales como, por ejemplo, las proporcionadas por IMS. La iniciativa SCORM pone en práctica diferentes desarrollos tecnológicos de las iniciativas propuestas por grupos como IMS, AICC, ARIADNE y IEEE-LTSC, todos ellos agrupados en un único modelo de referencia para especificar una implementación

consistente que pueda ser utilizada por toda la comunidad de e-learning. SCORM define los fundamentos técnicos de un LMS basado en tecnologías web, estableciendo:

- Un *modelo de agregación de contenidos* que describe los componentes utilizados dentro de una experiencia educativa, cómo empaquetar estos componentes para su intercambio, y cómo describir estos componentes mediante el uso de los metadatos para permitir su búsqueda y descubrimiento. Además también define los requisitos para la construcción de agregaciones de contenidos (v.g. cursos, lecciones, módulos, etc.). Este modelo da lugar a un concepto de LO denominado Shareable Content Object (SCO).
- Un *entorno de ejecución dinámico* para la reproducción de los SCOs, proporcionando de esta forma, un modelo instruccional adaptativo basado en LO. Normalmente dicho entorno de ejecución se servirá a los distintos clientes que acceden al sistema.
- Un *mecanismo de interoperabilidad* entre los SCOs que se ejecutan en el citado entorno y el resto de componentes que se ejecutan en el LMS que, habitualmente, reside en el lado del servidor.

Adicionalmente, en SCORM 2004 (anteriormente conocido como SCORM 1.3) se ha introducido un *Modelo de Secuenciación y Navegación* que permite la presentación dinámica de los contenidos educativos en función de las necesidades de aprendizaje. Está basado en la propuesta IMS Simple Sequencing (IMS-SS) y describe cómo se puede secuenciar el contenido compatible con el modelo SCORM utilizando una secuencia de eventos de navegación, donde estos eventos pueden ser iniciados por el estudiante o por el sistema. El control de las bifurcaciones y el flujo puede ser descrito utilizando un conjunto predefinido de acciones que normalmente se deben fijar durante el diseño del curso. Además, también describe cómo los LMSs compatibles con SCORM tienen que interpretar estas reglas de secuenciamiento expresadas por el desarrollador de contenidos, acompañadas por el conjunto de eventos de navegación iniciados por el estudiante o por el sistema, y cuáles son sus efectos sobre el entorno en tiempo de ejecución. SCORM utiliza el concepto de *organización de contenido* (en inglés *content organization*) como UoL.

Cabe destacar en la actualidad la iniciativa internacional LETSI (www.lets.org) que tiene como objetivo promover y desarrollar la siguiente generación de marcos de trabajo interoperable entre los sistemas educativos en base a la experiencia adquirida en la interoperabilidad con SCORM y otros marcos de trabajo similares. Con este objetivo LETSI ha formado varios grupos de trabajo para analizar diversos aspectos (v.g. secuenciamiento, arquitectura, intercambio de información, etc.) con el objetivo de promover una nueva versión de SCORM que denominan SCORM 2.0. El trabajo actual llevado a cabo en LETSI en línea con SCORM 2.0 (Barr, A., 2010)

(Barr, A., 2010b) pasa por integrar actividades educativas de diversos tipos adicionalmente a los contenidos educativos, actividades colaborativas y la integración de servicios externos distribuidos que proporcionen dichas actividades.

eLML (eLesson Markup Language) (Fisler et al., 2005) es un lenguaje específico del dominio basado en XML que permite crear lecciones estructuradas descritas como documentos XML (eLessons). Las lecciones así descritas se pueden transformar en múltiples formatos de salida, tales como IMS Content Package o SCORM, HTML, eBooks expresados en el formato ePub, PDF, etc.

<e-DocBook> es una metodología y un conjunto de herramientas ideadas para simplificar el proceso de creación de materiales educativos adaptativos basados en el concepto de LO (Martínez-Ortiz et al., 2005) (Martínez-Ortiz et al., 2006), que sigue un enfoque similar a eLML, pero adoptando tecnologías ya existentes. Efectivamente, <e-DocBook> utiliza una extensión de DocBook (Walsh & Muellner, 1999), un lenguaje XML usado en entornos técnicos para la creación de manuales. De esta forma, la metodología propuesta por <e-DocBook> se basa en la metáfora de escritura de un manual, proceso al que habitualmente están acostumbrados los docentes (aunque para e-learning los contenidos deberían ser concebidos de forma diferente, en base a los conceptos que se quieren enseñar y su relación con los LO, y no sólo como un libro que se publica en la red). A partir del manual generado, y aplicando las herramientas proporcionadas por <e-DocBook>, se pueden obtener distintos productos: un curso, módulos de curso basados en LO, transparencias para ser utilizadas como notas de clase, etc. En ambos casos el resultado final será agregado en un paquete IMS Content Packaging (IMS-CP) para poder simplificar el intercambio de los contenidos entre sistemas de e-learning. Además, las herramientas proporcionadas permiten la generación de los contenidos en distintos formatos como, por ejemplo, HTML, PS, PDF, RTF, archivos de ayuda de Eclipse o archivos de ayuda de Windows.

2.1.3.3. Lenguajes de Actividades

OUNL-EML fue desarrollado por la OUNL para su aplicación en e-learning. La versión 1.0 de dicho lenguaje y su modelo XML fue distribuida en el año 2000 (Koper, 2000) (Koper, 2001). OUNL-EML fue seleccionado como base de la especificación IMS-LD, integrando OUNL-EML con la especificación IMS-CP e IMS-SS. OUNL-EML ha sido utilizado y puesto en práctica en diversas aplicaciones de e-learning, y, en particular, en la Universidad Complutense de Madrid en las primeras versiones del proyecto <e-Aula> (Sierra et al., 2007b). OUNL-EML permite la definición de *diseños de aprendizaje* (diseños instructivos) con el objetivo de permitir la creación de herramientas avanzadas de e-learning (v.g. herramientas que permitan la definición de un modelo educacional basado en competencias, en portafolio o en el aprendizaje colaborativo). En OUNL-EML una UoL es el elemento central de un *diseño de aprendizaje* (*learning design*). Un diseño de aprendizaje es una instancia concreta de un modelo pedagógico, el cual a su vez es una

instancia de un metamodelo pedagógico. Este metamodelo no fuerza a los usuarios de OUNL-EML a utilizar un modelo pedagógico concreto, sino que les permite crear y describir sus propios modelos de manera expresiva. El metamodelo ofrecido por OUNL-EML ha sido desarrollado a partir del análisis de los modelos existentes basados en las aproximaciones constructivistas (sociales), de comportamiento y cognitivas.

PALO es un lenguaje de modelado desarrollado en la Universidad Nacional de Enseñanza a Distancia (UNED) (Rodríguez-Artacho et al., 1999). PALO permite describir cursos organizados mediante módulos que contienen actividades educativas, contenido y un plan de enseñanza. Utilizando PALO el diseñador puede crear plantillas para definir tipos de escenarios de aprendizaje. Utilizando las características del lenguaje, es posible secuenciar tareas de aprendizaje y módulos. Adicionalmente se pueden añadir restricciones sobre los cursos, permitiendo definir tiempos y fechas límite, así como dependencias entre módulos y tareas. Utiliza el concepto de módulo (*course*) como UoL.

Educational Environment Modeling Language (E2ML) es una propuesta de EML que proporciona un lenguaje visual con el objetivo de permitir diseñar escenarios educativos en el ámbito universitario (Botturi, 2006). Dicho lenguaje permite crear una definición explícita del proceso de aprendizaje y de las actividades educativas. En particular aborda los siguientes aspectos:

- Facilita la comunicación entre los diferentes interesados dentro del proceso (diseñadores de unidades de aprendizaje, personal técnico, profesores, etc.). Para ello propone una representación visual del diseño, que se puede utilizar de manera similar a como se utilizan los planos de un edificio que va a construir.
- El diseño de una UoL puede utilizarse como base de otra UoL, no sólo por el mismo diseñador, sino por el resto de la comunidad educativa.

E2ML ha sido ideado como una herramienta de comunicación, una notación que pueda ser utilizada por todos los expertos involucrados en el diseño, creación y puesta en práctica de diseños educativos. No obstante, en su versión más avanzada (Botturi et. al., 2006) E2ML es compatible con IMS-LD, de modo que los diseños creados con E2ML puedan traducirse a IMS-LD.

MISA introduce una nueva aproximación denominada *ingeniería instruccional* (en inglés *Instructional Engineering*) (Paquette, 2004). La Ingeniería Instruccional está basada en las teorías del *Diseño Instruccional* (del término en inglés *Instructional Design*) (Reigeluth, 1983) (Merrill, 1994) (Dick et al., 2000) junto a la ingeniería cognitiva y la ingeniería del software. Para ello proporciona una metodología que da soporte a la planificación, análisis, diseño y entrega de un sistema de aprendizaje y comparte los principios de los EMLs. MISA permite el diseño de sistemas de aprendizaje a través de 35 tareas, produciendo 35 entregables denominados elementos

de documentación. La creación de estos documentos está dividida en fases bien definidas. El concepto de *escenario de aprendizaje* representa una UoL en MISA.

XEDU está ideado para ofrecer a los diseñadores instruccionales un marco de trabajo para la especificación de cualquier aplicación educativa, tanto desde el punto de vista de las teorías instruccionales como desde el punto de vista de la ingeniería del software (Buendía-García & Díaz-Perez, 2003) (Buendía-García et al., 2004). Los principales conceptos definidos en XEDU son:

- *Perfil de alumno*. Almacena toda la información relevante acerca del estudiante, incluyendo el resultado del proceso de aprendizaje.
- *Escenario educativo*. Consta de actividades y condiciones en un contexto educativo específico.
- *Estructura didáctica*. Organiza el contenido educativo con un objetivo didáctico específico.

El concepto de *estructura didáctica* representa una UoL en XEDU.

<e-LD> es una iniciativa que trata de simplificar la autoría y reutilización de diseños educativos mediante la aplicación de conceptos de flujos de trabajo (en inglés *workflows*), así como de los principios de la Ingeniería de Lenguajes Software (Martínez-Ortiz et al., 2008). La idea es proponer un lenguaje específico de dominio orientado a flujo para el secuenciamiento de actividades, que es más comprensible para los docentes. La compatibilidad con otros estándares como IMS-LD se obtiene mediante procesos de exportación automática. Además proporciona herramientas para ayudar a la comprensión de diseños previamente creados con IMS-LD de los que únicamente se dispone de su representación en XML (permitiendo, por ejemplo, la visualización de dependencias entre tareas o entre tareas y condiciones).

Una iniciativa similar a <e-LD> es Flexo LD (Dodero et al., 2010). Al igual que <e-LD>, este lenguaje permite describir diseños educativos exportables a IMS-LD utilizando una notación gráfica. También ofrece una notación textual alternativa, basada en un lenguaje específico que permite descripciones menos tersas que las promovidas por la vinculación a XML de IMS-LD.

PoEML (Perspective-oriented EML) es un lenguaje creado a partir de un análisis crítico de IMS-LD que se define a partir de la propuesta de separación de asuntos o aspectos del proceso educativo (Caeiro et al., 2007). El lenguaje tiene una estructura modular con la que se pretende mejorar los problemas de capacidad expresiva, complejidad y usabilidad identificados en los EMLs actuales.

Learning Activity Management System (LAMS) es una herramienta que permite el diseño, gestión y distribución de secuencias de actividades colaborativas para e-learning (Dalziel, 2003) (Dalziel, 2005). LAMS proporciona una herramienta de autoría visual muy intuitiva (Fig. 2.1) que permite crear las secuencias de actividades de aprendizaje. En los cursos de LAMS se pueden

encontrar diferentes actividades, como actividades individuales, actividades para pequeños grupos de usuarios o para clases completas. Estas actividades pueden incluir trabajo con contenidos educativos o actividades de trabajo colaborativo (Fernández-Manjón et al., 2009). En su versión actual (2.3 a la fecha de escritura de esta Tesis) LAMS puede exportar sus secuencias de actividades como una UoL compatible con IMS-LD nivel A. Nótese que, debido a esta restricción, la UoL resultante pierde parte de la semántica de secuenciamiento de actividades. Para LAMS el concepto de UoL se denomina *secuencia de actividades*.

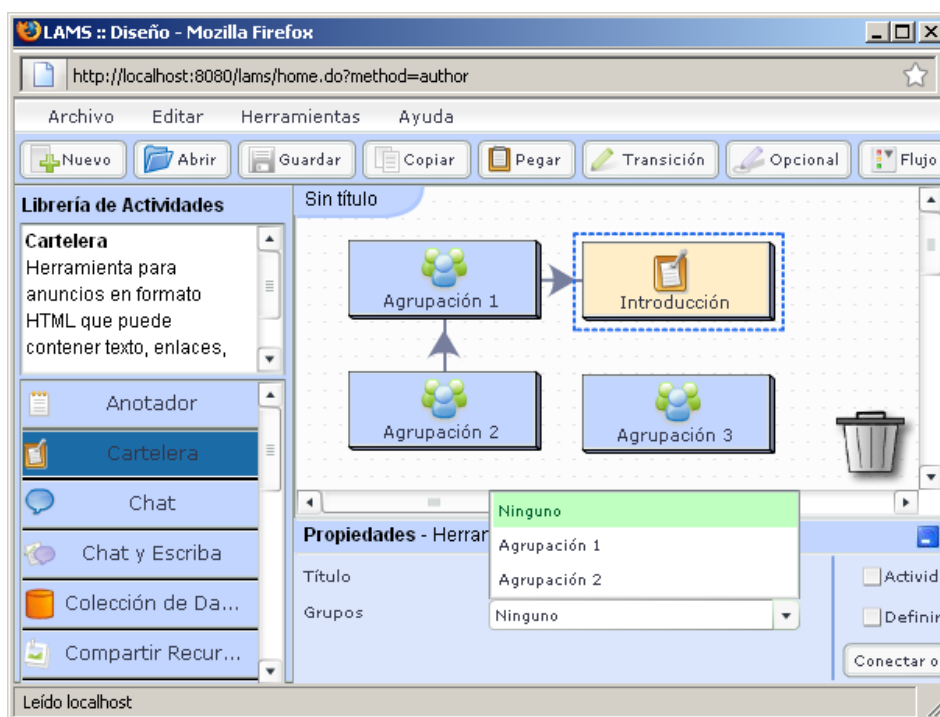


Fig. 2.1. Interfaz del editor de LAMS.

Learning Design Visual Sequence (LDVS) es una notación visual creada en el marco de un proyecto nacional Australiano con el objetivo de identificar y describir las prácticas educativas innovadoras empleando las TIC (Agostinho et. al., 2008). Este proyecto, conocido como *Learning Designs Project* (<http://www.learningdesigns.uow.edu.au>) recopila diseños educativos genéricos, recursos y otras herramientas con el propósito de ayudar a los profesores de educación universitaria a implementar diseños educativos innovadores que hagan uso de las TIC. En LDVS un diseño educativo está compuesto de tres elementos: las tareas que los estudiantes tienen que completar, los contenidos educativos proporcionados a los alumnos para llevar a cabo las tareas, y actividades de apoyo para el guiado de los estudiantes en el proceso de aprendizaje (Agostinho, 2009). Los diagramas creados con la notación LDVS muestran la cronología de los elementos que forman el lenguaje asignando a cada uno de los tres elementos una notación gráfica diferente. En

resumen, un diseño educativo en LDVS está constituido por un diagrama, que utiliza la notación LDVS, que representa un resumen visual del diseño educativo desde la perspectiva del profesor, y un documento (textual) detallado que explica cómo las tareas, recursos y actividades de apoyo han sido diseñadas e implementadas, además de describir el contexto educativo en el que han sido puestas en práctica.

Learning Design Language (LDL) (Martel et al., 2006) (Martel et al., 2006b) (Ferraris et al., 2008) (Ferraris et al., 2008b) es un EML diseñado especialmente para modelar actividades colaborativas. La definición de una UoL se denomina *escenario* en LDL, definiéndose como una descripción formal de una actividad futura. Un *escenario* LDL está compuesto por:

- *Estructuras*. Representan los elementos de secuenciamiento implícito del lenguaje. En LDL existen 3 tipos de estructuras: secuencia, selección y aleatoria, que permiten organizar el orden en el que las interacciones se ejecutan.
- *Roles*. Representan los papeles de los participantes que estarán involucrados en el escenario.
- *Reglas*. Definen las condiciones utilizadas para controlar el comienzo y la finalización de las interacciones y estructuras dentro de la actividad.
- *Observables*. En LDL todos los conceptos son observables, permitiendo la posibilidad de definir distintos puntos de observación en cada elemento observable.
- *Interacciones*. Representan las interacciones entre los participantes del escenario.

Los diseños creados utilizando LDL pueden ser automatizados utilizando una herramienta, creada por los mismos autores de LDL, denominada Learning Design Infrastructure (LDI) (Martel et al., 2006).

Entre las diferentes propuestas de especificaciones, IMS Learning Design ha emergido como el estándar de facto para la representación de cualquier UoL, ya que, en principio, permite utilizar cualquier teoría de aprendizaje. Debido a su importancia tanto en el ámbito del e-learning como para este trabajo, se realiza una presentación breve de esta especificación en la siguiente sección.

2.1.4. IMS Learning Design

IMS Learning Design está basado en el lenguaje OUNL-EML. Esta especificación ha sido elaborada por el grupo de trabajo IMS/LDWG, enmarcado dentro de las iniciativas desarrolladas por el IMS Global Learning Consortium (IMS, 2003), tomando como punto de partida la propuesta OUNL-EML. El resultado final ha sido el desarrollo de una nueva especificación adaptada en aquellas partes en las que existía un solapamiento con el resto de especificaciones propuestas por

IMS. La especificación IMS-LD está organizada en varios niveles, al estilo del modelo de capas en una arquitectura de software, para hacerla más comprensible.

Una de las adaptaciones más relevantes llevada a cabo ha sido la adopción de la especificación IMS Content Packaging como formato y medio de transmisión e intercambio entre distintos LMSs y herramientas. Así mismo, otras partes definidas en OUNL-EML no han sido reutilizadas, por ejemplo, el formato XML para la descripción de los contenidos educativos, que era un dialecto del formato DocBook (Walsh & Muellner, 1999).

Además de la adaptación al entorno de especificaciones de IMS, otro objetivo ha sido la inclusión de otros trabajos y especificaciones que no se solapaban con el trabajo realizado. Por ejemplo, IMS-SS ha sido incluido dentro del ámbito de trabajo de IMS Learning Design para llevar a cabo el secuenciamiento de los LO dentro de las actividades que se llevan a cabo en el contexto de la UoL. Otro ejemplo ha sido IMS Question and Test Interoperability, de manera que los exámenes y preguntas presentadas con IMS QTI pueden utilizarse dentro de las actividades. Así mismo, la puntuación obtenida en estas pruebas puede provocar que aparezcan nuevas actividades o que desaparezcan otras.

En la Fig. 2.2 puede observarse la estructura de alto nivel de una UoL expresada en IMS-LD.

Además, para facilitar el aprendizaje y la implementación de la especificación, el modelo conceptual de IMS-LD (ver Fig. 2.3) está dividido en tres niveles (A, B y C) donde cada nivel está construido encima del modelo y de la semántica definida en los niveles previos:

- Nivel A. Contiene el núcleo de los componentes de la especificación. Cuando se crea una UoL con IMS-LD, se debe especificar un modelo estático y un modelo dinámico.
- Nivel B. Añade los conceptos de propiedades y condiciones al nivel anterior. Las propiedades definen un modelo de datos para el alumno, y las condiciones se utilizan para personalizar las UoLs en base a los conocimientos previos de los alumnos y a su interacción con dichas UoLs.
- Nivel C. Añade el concepto de notificación al nivel anterior. Las notificaciones proporcionan un nuevo mecanismo de notificación de sucesos que ocurren durante la ejecución de una UoL.

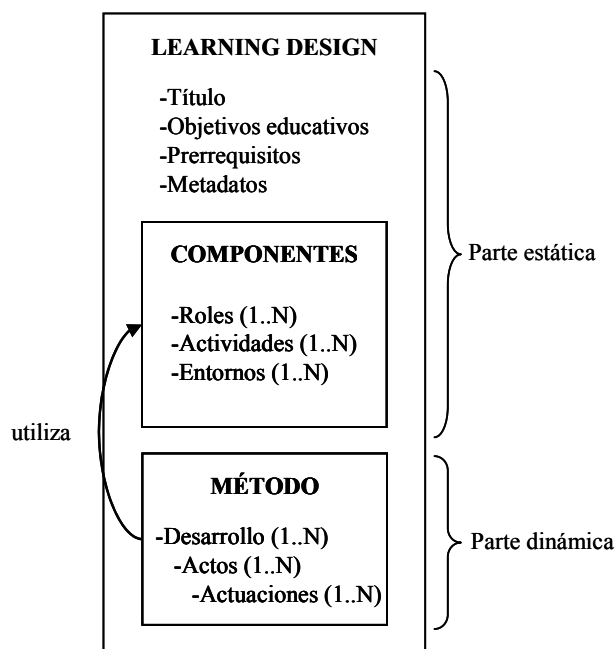


Fig. 2.2. Estructura de alto nivel de una unidad de aprendizaje (UoL) en IMS-LD.

El modelo estático de IMS-LD permite definir qué es lo que se va llevar a cabo en la UoL, qué tipos de usuarios (v.g. profesores, alumnos, etc.) participan en la UoL, y con qué recursos se llevarán a cabo las actividades (v.g. páginas HTML, PDF, etc.). El modelo estático está compuesto por los siguientes conceptos:

- *Título*. Permite dar un título a la UoL.
- *Objetivos educativos*. Definición de los objetivos educativos de la UoL. Esta descripción habitualmente se realiza en lenguaje natural.
- *Prerrequisitos*. Definición de los conocimientos previos que debe tener un alumno para que pueda llevar a cabo las actividades de la UoL. La descripción de los prerrequisitos también se realiza en lenguaje natural.
- *Metadatos*. Esta metainformación permite la indexación de las UoLs, de manera que se simplifique su posterior clasificación, catalogación y recuperación.
- *Roles*. Definición los diferentes tipos de usuario que participarán en la UoL.
- *Actividades*. Definición de las actividades que se llevarán a cabo dentro de las UoLs. En la definición de las actividades se incluyen los objetivos y los prerrequisitos de las actividades. Además, también se incluye una descripción textual de cuáles son los objetivos de la actividad. Finalmente, se incluye una referencia al entorno donde se llevará a cabo la actividad.

- *Entornos*. Definen los contenidos educativos (LO) y las herramientas (servicios de aprendizaje) que se utilizarán en las distintas actividades de la UoL.

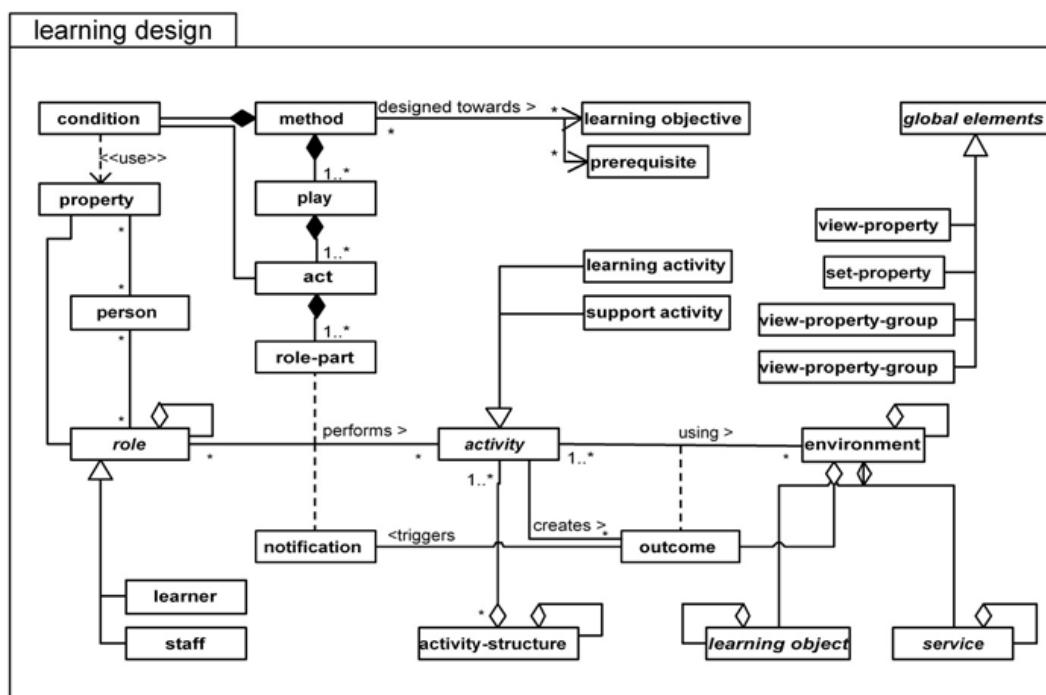


Fig. 2.3. Estructura conceptual de IMS-LD (tomada de IMS, 2003)

El modelo dinámico de IMS-LD permite definir el quién y el cuándo. Para ello, hace uso de los conceptos definidos en el modelo estático anteriormente descrito. De esta forma, el modelo dinámico permite definir qué tipo de usuario (rol) llevará a cabo cada actividad concreta y también permite definir la sincronización y las dependencias que existirán entre las distintas actividades que componen la UoL.

El modelo dinámico puede verse como la escenificación de una obra teatral. El método (method) consiste en una o varias representaciones (play) que son interpretadas en paralelo. Cada una de las obras está formada por uno o más actos (acts) que serán interpretados uno tras otro (ver Fig. 2.4).

Dentro de cada acto se realiza la distribución de papeles (role-parts), es decir, se define qué actividades serán llevadas a cabo por cada uno de los tipos de usuario involucrados en el proceso de aprendizaje (ver Fig. 2.4).

Utilizando los niveles B y C podemos crear secuenciamientos dinámicos más complejos. En particular se pueden incluir dependencias entre actividades (por ejemplo, para indicar que un profesor debe desempeñar una actividad en la que corrija un examen después de que el alumno lo termina).

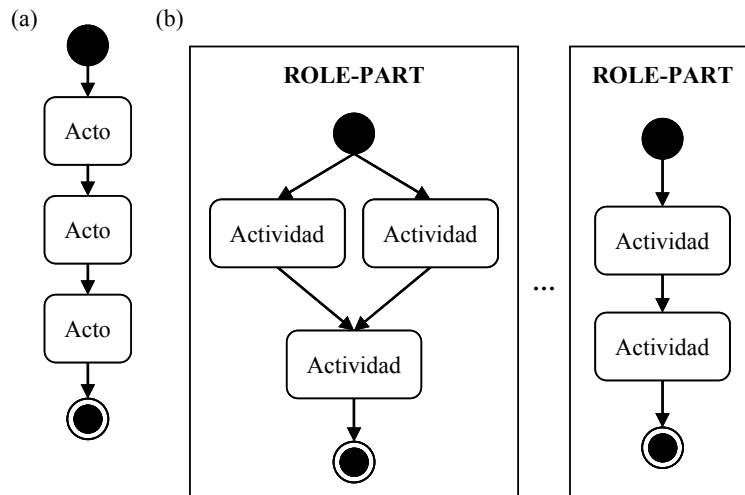


Fig. 2.4. Representación de un método con tres actos secuenciales (a) y representación de un método en el que se incluyen los roles de los usuarios en las actividades y se muestra que puede haber actividades en paralelo (b).

2.1.5. Herramientas de soporte a IMS-LD

Aunque la aplicación práctica de IMS-LD sigue siendo muy limitada, sí existen diversas iniciativas que proporcionan herramientas para trabajar con IMS-LD.

En esta sección se realiza una breve presentación de algunas de las iniciativas que, debido a su relevancia o madurez, se han considerado como las más prometedoras actualmente.

Podemos encontrar tres tipos de herramientas:

- *Herramientas de Autoría.* Son herramientas que permiten la creación de las UoLs.
- *Motores de Ejecución.* Son herramientas que, dada una UoL codificada con IMS-LD, interpretan el proceso de aprendizaje, monitorizando la realización de las actividades y actualizando el perfil del alumno según los resultados que se vayan obteniendo en las actividades que tiene asignadas. Este tipo de herramientas habitualmente residen en un servidor de aplicaciones, y son utilizadas tanto por los profesores como por los alumnos a través de una interfaz web adecuada.
- *Reproductores.* Estas herramientas son utilizadas por los actores que interactúan con la UoL, tanto en el proceso de ejecución de la misma, como durante el proceso de publicación de la UoL, es decir, el proceso de preparación de la misma para su puesta en ejecución. De esta forma, estas herramientas proporcionan la citada interfaz web con los motores de ejecución.

2.1.5.1. Herramientas de Autoría de IMS-LD

En esta sección se hace un breve análisis de distintas iniciativas de desarrollo de herramientas de autoría compatibles con la especificación IMS-LD. Esta sección no pretende realizar un análisis exhaustivo de todas las iniciativas existentes (para mayor detalle se puede consultar (Berggreen et al., 2005), (Burgos & Griffiths, 2005), (Dodero et al., 2006) (Berlanga & García, 2005)), sino de las que en el momento de escritura del documento de esta Tesis se han considerado más relevantes y representativas: ALFANET, CopperAuthor, RELOAD, CoSMoS, Collage, MOT+ y ASK-LDT.

ALFANET (Active Learning for Adaptive Internet) es una iniciativa europea que tiene como objetivo el desarrollo de nuevos métodos y servicios para llevar a cabo un proceso de aprendizaje activo y adaptado al alumno (<http://adenu.ia.uned.es/alfanet/>). El editor de IMS-LD de ALFANET (<http://sourceforge.net/projects/alfanetat/>) representa los conceptos de IMS-LD mediante elementos gráficos que conforman la interfaz. Esta herramienta sólo permite la creación y el desarrollo de diseños educativos IMS-LD de nivel A (este proyecto ya está concluido y no está clara la continuidad de su desarrollo).

CopperAuthor (Fig. 2.5) es una herramienta desarrollada en paralelo al motor de ejecución CopperCore. Esta herramienta permite a los diseñadores construir y navegar sobre la estructura del diseño educativo mediante una interfaz basada en tablas (<http://sf.net/projects/copperauthor>). En su versión actual la interfaz es un tanto primitiva y sólo permite el desarrollo de diseños educativos IMS-LD de nivel A.

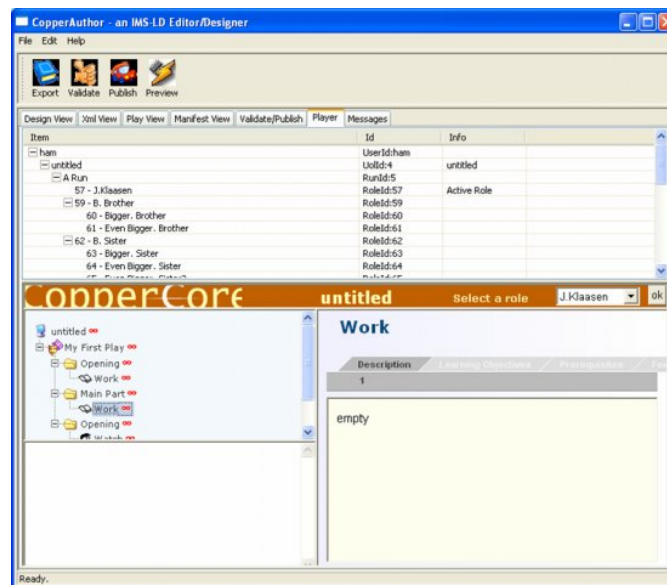


Fig. 2.5. Interfaz de CopperAuthor.

RELOAD (Reusable E-Learning Object Authoring & Delivery) (Fig. 2.6) es un editor desarrollado en el seno de un proyecto patrocinado por la iniciativa JISC del Reino Unido (<http://www.reload.ac.uk>). Este editor permite la edición de una UoL mediante la interacción con múltiples formularios y estructuras en forma de árbol que representan la estructura de agregación de los conceptos de IMS-LD implícita en el formato XML utilizado para representar las UoL de IMS-LD. Con esta herramienta se pueden crear diseños educativos IMS-LD de los tres niveles (del nivel A al nivel C).

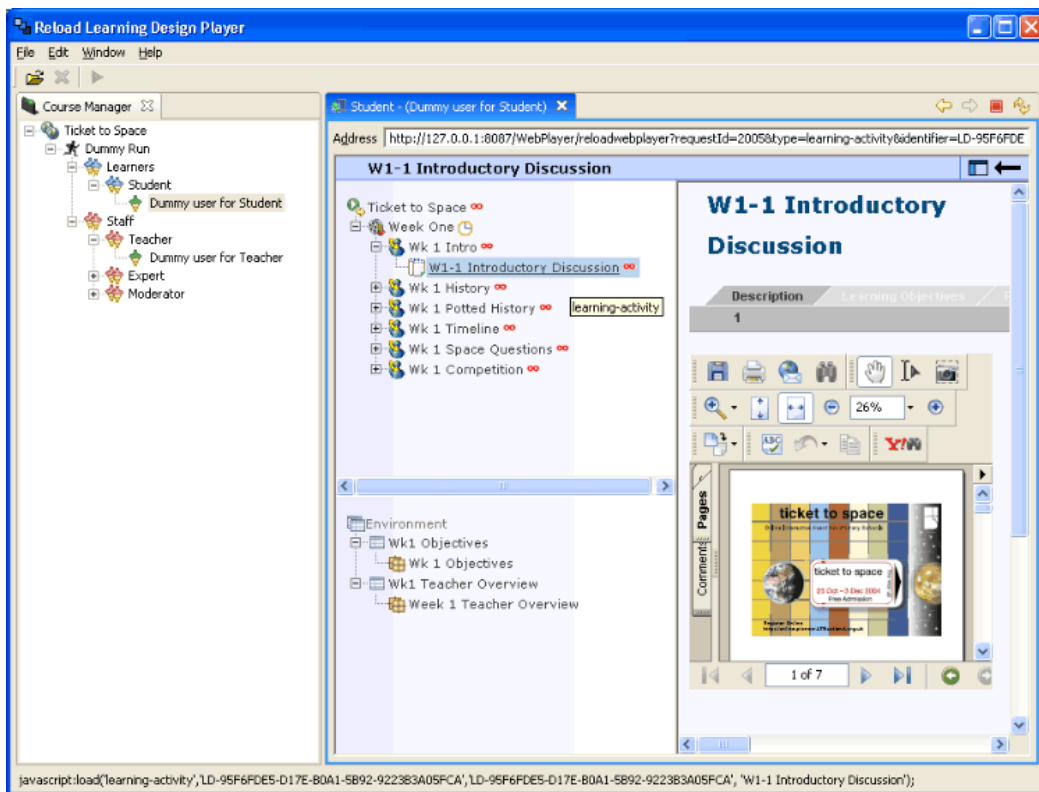


Fig. 2.6. Interfaz del editor de IMS-LD de Reload.

CoSMoS (Collaboration Script Modelling System) (Fig. 2.7) es una herramienta de autoría ideada inicialmente para dar soporte a la formalización de procesos de aprendizaje colaborativos (Miao, 2005). Posteriormente, la herramienta fue modificada para dar soporte a los conceptos de IMS-LD. La edición de la UoL se basa en la navegación sobre la estructura en árbol de la misma y la edición de los conceptos mediante formularios. Con esta herramienta se pueden crear diseños educativos IMS-LD de nivel B.

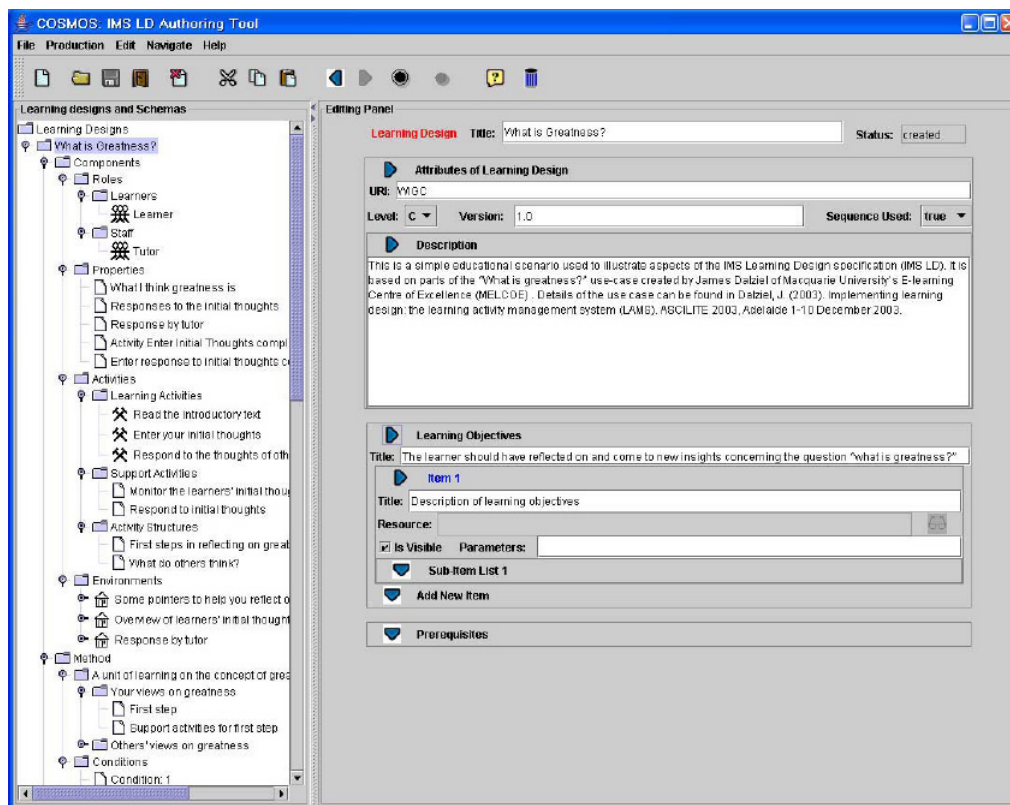


Fig. 2.7. Interfaz del editor de IMS-LD CoSMoS.

Collage es una herramienta de autoría de alto nivel y de autoría colaborativa basada en el concepto de los patrones de flujo colaborativos (Collaborative Learning Flow Patterns), que son plantillas que definen el flujo de tareas para dirigir de manera adecuada el proceso de aprendizaje (Hernández-Leo et. al., 2006). Esta herramienta ha sido desarrollada sobre RELOAD y con ella sólo se pueden crear diseños educativos IMS-LD de nivel A.

MOT+ es una herramienta desarrollada en el centro de investigación LICEF de Canadá, con el objetivo inicial de estructurar mapas conceptuales para la representación de conocimiento en diversos dominios (Paquette et al., 2005). MOT+ utiliza una notación gráfica para representar las entidades de conocimiento con las que trabaja la herramienta. MOT+ fue extendida para soportar la edición de IMS-LD, de manera que la herramienta permite representar y editar los conceptos que están definidos en IMS-LD. Con esta herramienta se pueden crear diseños educativos IMS-LD de nivel A y están desarrollando una extensión para soportar los niveles B y C.

ASK-LDT (Advanced e-Services for the Knowledge Society Research Unit, Fig. 2.8) es una herramienta que proporciona una notación gráfica para los conceptos de IMS-LD (Karampiperis & Sampson, 2004). ASK-LDT define una notación gráfica para un conjunto de tipos de actividades predefinidas, como, por ejemplo, lecciones, discusiones, exámenes, etc. Adicionalmente también

proporciona una notación gráfica para poder definir el flujo entre dichas actividades. Con esta herramienta se pueden crear diseños educativos IMS-LD de nivel A y parcialmente de nivel B.

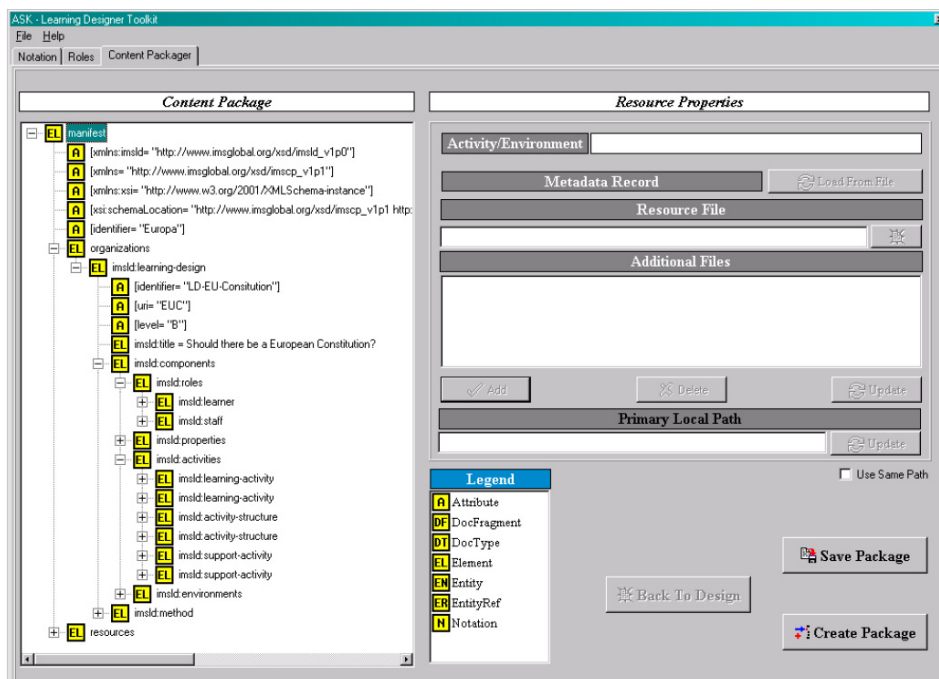


Fig. 2.8. Interfaz del editor ASK-LDT.

2.1.6. Motores de ejecución compatibles con IMS-LD

Quizá el motor IMS-LD más popular sea CopperCore. CopperCore es un motor de ejecución de UoLs formalizadas con IMS-LD desarrollado por la OUNL. Como características principales de CopperCore podemos destacar:

- Es un proyecto de software libre.
- Soporta la ejecución de UoLs hasta el nivel C de IMS-LD.
- CopperCore está desarrollado sobre la plataforma Java EE, y su puesta en funcionamiento es relativamente simple.
- CopperCore proporciona una Interfaz de Programación de Aplicaciones (API, del término en inglés *Application Programming Interface*) que permite extender sus funcionalidades, así como controlar el motor de ejecución desde otra herramienta.

Proporciona una capa de abstracción, CopperCore Service Integration (CCSI), que permite integrar de forma sencilla diversos servicios de aprendizaje como, por ejemplo, herramientas compatibles con IMS-QTI.

Otro entorno de ejecución de IMS-LD es GRAIL (Gradient-lab RTE for Adaptive IMS-LD in .LRN) desarrollado en la Universidad Carlos III de Madrid que permite la ejecución de UoL en el LMS de código libre .LRN (Escobedo et al., 2007). GRAIL ejecuta UoLs de los tres niveles que permite la especificación y es una de las pocas herramientas disponibles que se integra completamente en un LMS ampliamente difundido y utilizado (.LRN).

Finalmente, existe otra iniciativa denominada OPENET LD (Vidal, J. C. et al., 2009) que tiene como objetivo crear un motor de ejecución de IMS-LD en el que la semántica de IMS-LD se especifica mediante el uso de ontologías (Gómez-Pérez, A., et al., 2004) y la ejecución de la UoL se lleva a cabo utilizando redes de Petri (Murata, 1989). OPENET LD ejecuta UoLs de los niveles A y B de la especificación.

2.1.7. Reproductores de IMS-LD

CopperCore Player (Fig. 2.9) es una aplicación web simple que permite interactuar con el motor de ejecución CopperCore. Esta herramienta fue creada como una herramienta simple para realizar las pruebas necesarias durante el desarrollo del motor CopperCore.

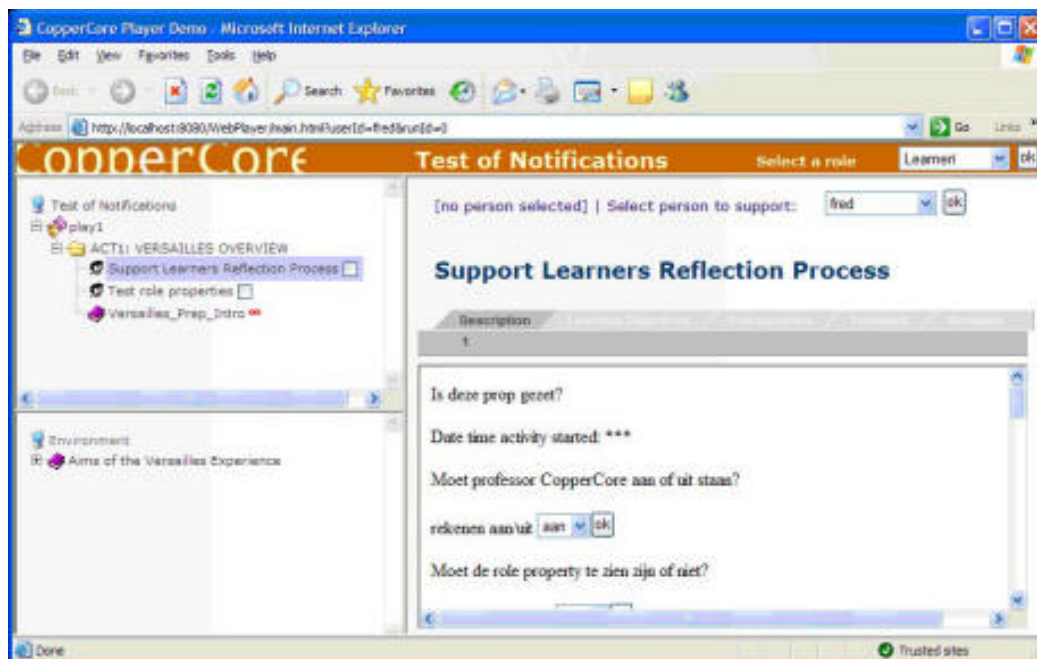


Fig. 2.9. Interfaz web del reproductor Coppercore web player.

SLeD Player (Fig. 2.10) es un nuevo cliente web para el motor de ejecución CopperCore (McAndrew et al., 2004) (Weller et. al., 2006). Las principales características de SLeD Player son:

- Proporciona una interfaz web para la gestión de usuarios y de las ejecuciones de las UoLs.

- Permite la personalización del diseño y de la distribución de la interfaz de reproducción de la UoL mediante el uso de descriptores XML.
- Proporciona implementaciones para los servicios de búsqueda y foro que pueden ser referenciados dentro de los entornos de una UoL codificada con IMS-LD.
- Proporciona soporte a la capa de abstracción CopperCore Service Integration (CCSI) que tiene como objetivo integrar nuevos servicios educativos que puedan ser utilizados en colaboración con CopperCore. Como ejemplo de uso de CCSI, SLeD integra los servicios de foro y de búsqueda.

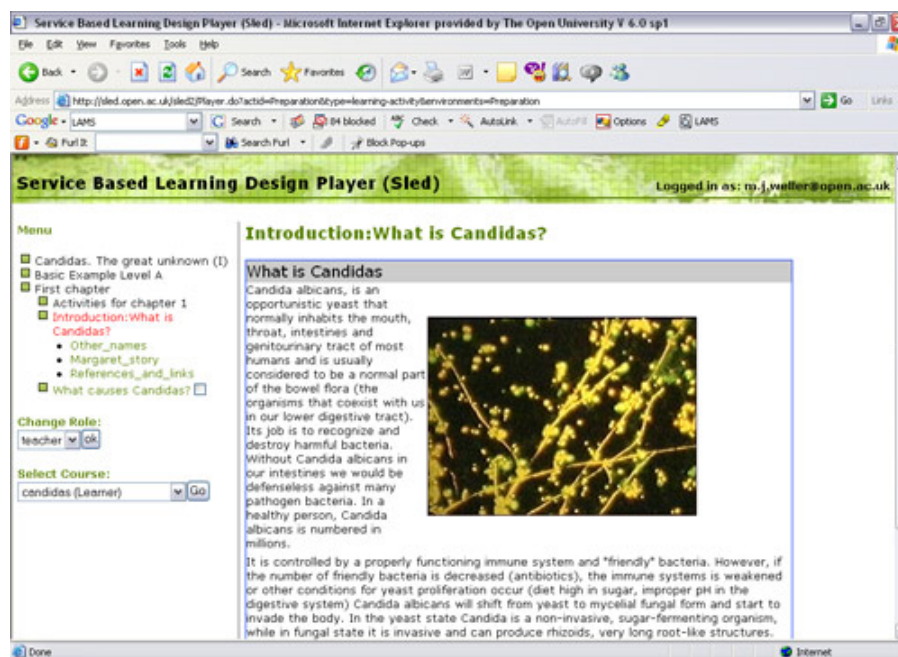


Fig. 2.10. Interfaz web del reproductor SLeD Player.

Reload Player (Fig. 2.11) ha sido desarrollado por Paul Sharples y Phillip Beauvoir en la Universidad de Bolton. Reload Player ha sido construido sobre la plataforma Eclipse y hace uso de Coppercore como motor de ejecución de IMS-LD. Esta herramienta proporciona una interfaz simple para la publicación de UoLs compatibles con IMS-LD y la creación de usuarios de prueba que pueden ser utilizados para probar las UoLs. Reload Player está pensada principalmente para probar de manera simple las UoLs que se están diseñando con el correspondiente editor.

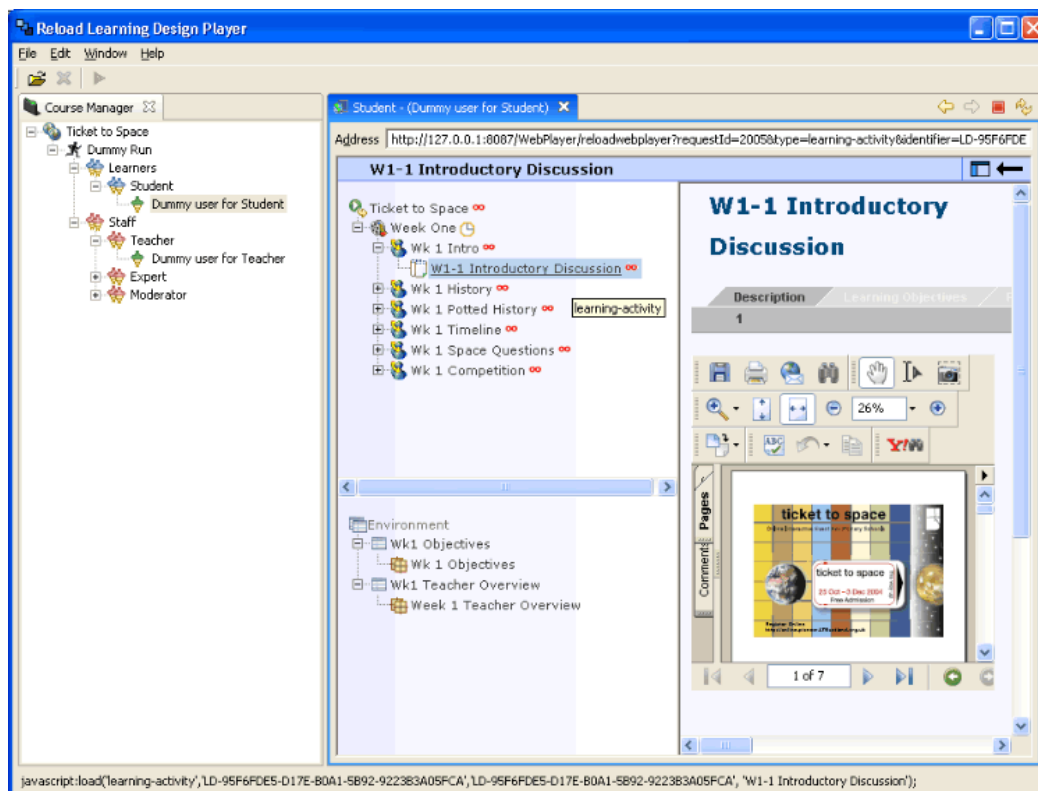


Fig. 2.11. Interfaz del reproductor de IMS-LD Reload Player.

2.1.8. Otras iniciativas y proyectos de investigación relacionados con IMS-LD

Como este campo del modelado educativo no está suficientemente maduro para su aplicación industrial y generalizada, una forma de mantenerse al corriente de los avances es considerar los proyectos de investigación relacionados. Aunque los proyectos que tratan, al menos en parte, estos aspectos son muy numerosos, nos quedaremos con algunos que, por su volumen y número de socios implicados, son susceptibles de tener un cierto impacto, bien en la evolución de las especificaciones o bien en el desarrollo de herramientas.

Prolix y TenCompetence son dos proyectos europeos desarrollados en el Sexto Programa Marco que están relacionados con IMS-LD y que se plantearon como objetivos avanzar en el uso de dicha especificación y, hasta cierto punto, producir herramientas que simplifiquen su aplicación. En Prolix (<http://www.prolixproject.org>) se creó una herramienta de autoría denominada Prolix LD authoring tool (Heyer et. al., 2007) que traduce un diseño de alto nivel a IMS-LD. TenCompetence es otro proyecto europeo muy relacionado con el modelado educativo y las competencias (<http://www.tencompetence.org/>), donde el mismo equipo que participó en el

Otro sitio web donde se puede encontrar mucha información relacionada con e-learning y el modelado educativo, tanto en sus aspectos más técnicos como en sus aspectos más educativos, es la Edutech wiki (http://edutechwiki.unige.ch/en/Main_Page) mantenida en la Universidad de Ginebra.

2.2. Lenguajes de Procesos de Negocio

2.2.1. Lenguajes de Modelado Educativo y Gestión de Procesos de Negocio

Los Lenguajes de Modelado Educativo dentro del campo del e-learning son relativamente recientes y, por tanto, sólo se dispone de una limitada experiencia de uso. No obstante, estos lenguajes pueden considerarse como una particularización al campo educativo de los lenguajes de modelado de negocios. En nuestra opinión, esta analogía puede permitir aprovechar tanto la experiencia existente en el campo del modelado y automatización de los negocios como el conjunto de técnicas y herramientas de soporte que se han desarrollado previamente.

La gestión de los procesos de negocio, debido a su interés económico, es un tema muy estudiado desde los comienzos de la revolución industrial, de modo que ha sido descrito desde cada uno de los posibles puntos de vista: económico, sociológico, psicológico, contable, empresarial y de ingeniería. En una primera aproximación, un proceso de negocio puede verse simplemente como una colección de actividades que son críticas para las funciones y servicios que proporciona una empresa.

En la actualidad, la aplicación de las Tecnologías de la Información y Comunicación ha provocado un gran avance en la organización, sistematización y automatización de los procesos de negocio. Este avance se ha visto reflejado en el desarrollo de aplicaciones software genéricas para la gestión de procesos de negocio que se engloban dentro del término Sistemas de Gestión de Flujo de Trabajo (WFMSs, del término en inglés *Workflow Management Systems*).

Básicamente, un flujo de trabajo consiste en la coordinación de un conjunto de actividades que se llevan a cabo para lograr un objetivo. La gestión de los flujos de trabajo tiene como objetivo proporcionar el soporte de secuenciamiento de las actividades (i.e. del flujo de trabajo) en una organización, de manera que el trabajo sea llevado a cabo de manera eficiente, por la persona adecuada y utilizando la herramienta adecuada. La analogía con los procesos de ejecución de los EMLs orientados a la descripción de actividades es, de esta forma, evidente.

2.2.2. Lenguajes de Descripción de Procesos de Negocio

Los WFMSs son las herramientas que proporcionan la infraestructura necesaria para ejecutar los procesos de negocio. Por tanto, es necesario formalizar de alguna manera los procesos de

metamodelo para la descripción del proceso de negocio y proporciona una vinculación con XML para permitir el intercambio de los procesos. La especificación XPDL utiliza un lenguaje XML como notación para el intercambio de definiciones de los procesos. De esta manera, los distintos productos y herramientas pueden mantener su representación interna para la definición de procesos, y a la vez importar/exportar las definiciones de proceso de una manera estandarizada. En la Fig. 2.13 se muestra el metamodelo de XPDL, metamodelo que incluye el conjunto básico de entidades que, según esta especificación, es necesario especificar en la descripción de un proceso de negocio. Para un mayor detalle sobre esta especificación puede consultarse (WfMC, 2002).

2.2.2.2. Web Services Business Process Execution Language (WS-BPEL)

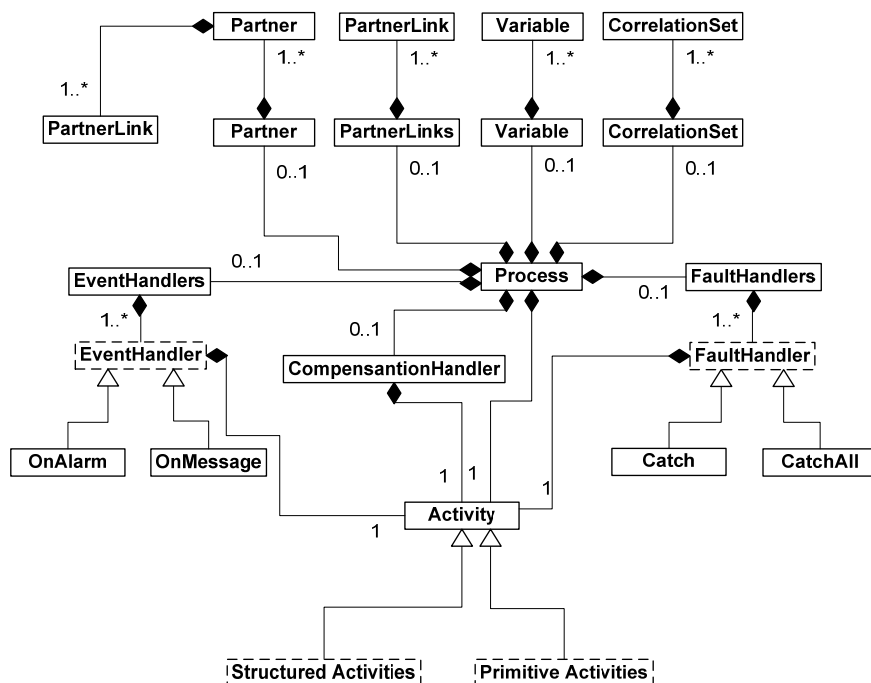


Fig. 2.14. Metamodelo que incluye los conceptos básicos de WS-BPEL.

WS-BPEL (Business Process Execution Language for Web Services) es un lenguaje XML diseñado para permitir la realización de tareas distribuidas, incluso entre múltiples organizaciones, utilizando una combinación de servicios web (Papzoglou & Georgakopoulos, 2003) (Cerami, 2002). WS-BPEL es por tanto un lenguaje basado en la composición de servicios web. De acuerdo con este lenguaje, los flujos de negocio se conciben como orquestaciones de servicios web (Peltz, 2003), permitiendo definir cómo deben ejecutarse un conjunto de servicios web básicos para dar cabida a las necesidades de los flujos de trabajo. Hay que destacar que un proceso WS-BPEL, a su vez, también se expone como un servicio web, con mensajes de entrada y opcionalmente respuestas vinculadas a operaciones WSDL ofrecidas por el proceso. La posibilidad de utilizar los

procesos WS-BPEL como servicios web permite la interacción entre distintos procesos, permitiendo la reutilización mediante composición y mejorando la escalabilidad. La Fig. 2.14 resume la estructura de la especificación WS-BPEL. Para un mayor detalle sobre esta especificación puede consultarse (Barreto et al., 2007) (Alves et al., 2007).

2.3. Ingeniería de Lenguajes Software

2.3.1. Lenguajes de Modelado Educativo e Ingeniería de Lenguajes Software

Tal y como se ha indicado repetidamente a lo largo de este capítulo, los EMLs pueden concebirse como lenguajes específicos de dominio que abordan la descripción y el desarrollo de distintos aspectos de un sistema e-learning. De esta forma, los principios básicos del desarrollo de software dirigido por lenguajes, la Ingeniería de los Lenguajes Software, están directamente relacionados con el diseño, implementación y uso de EMLs.

La Ingeniería de Lenguajes Software (Kleppe, 2008) promueve un desarrollo de software centrado en la formulación, implementación y uso de lenguajes específicos para cada dominio de aplicación: lenguajes específicos de dominio (DSLs, del inglés *Domain Specific Languages*). Bajo esta perspectiva, las aplicaciones se describen utilizando DSLs, y se generan o ejecutan automáticamente a partir de estas descripciones utilizando los procesadores (compiladores o intérpretes) asociados a dichos lenguajes. Dado que cada DSL incluye primitivas, medios de combinación y medios de abstracción cercanos a los conceptos y procesos utilizados en el dominio, las actividades de producción y mantenimiento de aplicaciones en dicho dominio se facilita enormemente. En el caso más extremo, serán los propios expertos en el dominio los que, equipados con el lenguaje específico, llevarán a cabo dicha producción y mantenimiento. Por su parte, los desarrolladores informáticos se situarán a un nivel metalingüístico superior, estando encargados de producir y mantener lenguajes específicos y procesadores para dichos lenguajes, así como herramientas que faciliten dichas actividades. De esta forma, en un escenario típico de desarrollo de software dirigido por lenguajes, es posible distinguir los siguientes roles (Deursen et al., 2000) (Kleppe, 2008) (Mernik et al., 2005):

- *Usuarios finales*: Son las personas que utilizan las aplicaciones producidas mediante los lenguajes específicos. En el caso de los EMLs, los usuarios finales serán los diferentes participantes en el proceso de aprendizaje (alumnos, pero también profesores, tutores, y demás personal de apoyo al proceso).
- *Usuarios de los lenguajes*: Son las personas que usan los lenguajes para construir las aplicaciones. Tal y como ya se ha indicado, en el caso más favorable, dicho rol recaerá en los

propios expertos del dominio (los instructores, en el caso de los EMLs), aunque en casos más realistas también se verán involucrados expertos en informática, que colaborarán estrechamente con los expertos del dominio en la construcción de las aplicaciones.

- *Proveedores de los lenguajes*: Son las personas que formulan e implementan los lenguajes específicos. Normalmente este rol recaerá en personas con experiencia en informática, desarrollo de software en general y en la definición e implementación de lenguajes informáticos en particular. No obstante, el alto grado de competencia exigida a estos participantes puede ser sustancialmente aliviado mediante el uso de herramientas apropiadas, que faciliten la formulación e implementación de los lenguajes.
- *Proveedores de las herramientas*: Son las personas que proporcionan las herramientas utilizadas en la definición, implementación y mantenimiento de los lenguajes. Éste es un rol altamente especializado, aunque también un rol distintivo en la Ingeniería de Lenguajes Software.

Tradicionalmente, la Ingeniería de Lenguajes Software se ha asociado con campos tales como la construcción de procesadores de lenguaje (compiladores e intérpretes) (Aho et al., 2006), los enfoques generativos a la reutilización de software (Krueger, 1992), los generadores de aplicaciones (Cleaveland, 1988) (Cleaveland, 2001), la programación generativa (Czarnecki, 2000), la definición de lenguajes visuales (Marriott et al., 1999), etc. En los últimos años la disciplina ha cobrado un énfasis especial con los métodos de desarrollo de software dirigido por modelos (Stahl et al., 2006), así como con la aparición de múltiples entornos de trabajo orientados a lenguajes (*language workbenches*) (Fowler, 2010), tales como Eclipse (Gronback, 2009) o Microsoft DSL Tools (Cook et al., 2007). Estos enfoques y tecnologías facilitan la especificación de modelos de muy alto nivel mediante lenguajes de modelado visuales específicos, así como la automatización de buena parte (o, incluso la totalidad) de la producción de una aplicación a partir de tales modelos. Estos avances han llevado a la moderna industria de desarrollo de software a hacerse eco de estas propuestas, considerándolas un elemento clave para la innovación o la mejora de sus procesos y la calidad de sus productos.

A continuación se examinan con mayor detalle los aspectos concretos relativos a los DSLs y al desarrollo de software dirigido por modelos.

2.3.2. Lenguajes Específicos de Dominio

El enfoque al desarrollo de software basado en DSLs tiene una larga tradición en informática, tal y como se evidencia en (Cleaveland, 1988) (Hudak, 1998) (Thibault et al. 1999) (Deursen et al., 2000) (Mernik et al., 2005). Cabe destacar que la mayor parte de estos trabajos se centran en

lenguajes de carácter textual, aunque como se indicará en la próxima sección, y siguiendo la tendencia actual en el campo, los principios subyacentes a dichos trabajos son fácilmente extrapolables al campo de los lenguajes gráficos y visuales.

Siguiendo a (Deursen et al., 2000), es posible definir un DSL como:

“Un lenguaje de programación o de especificación ejecutable que proporciona, mediante el uso de notaciones y abstracciones apropiadas, poder expresivo enfocado, y normalmente restringido, a un dominio de aplicación particular.”

De esta forma, y de acuerdo con (Deursen et al., 2000), las principales características de los DSLs son:

- Suelen ser lenguajes *pequeños* que, normalmente, ofrecen únicamente un repertorio restringido de notaciones y abstracciones. Es por ello que, habitualmente, se les denomina *micro-lenguajes* o *pequeños lenguajes* (del inglés, *Little Languages*) (Bentley, 1986). En ocasiones, no obstante, contienen un lenguaje de propósito general como sublenguaje, a fin de permitir extensibilidad de las aplicaciones que pueden describirse en términos específicos del dominio. Esta situación ocurre, normalmente, cuando los DSLs se implementan como *lenguajes embebidos* en un lenguaje de propósito general (Hudak, 1998).
- Suelen ser lenguajes *declarativos*. De esta forma, dichos lenguajes pueden verse, aparte de como lenguajes de programación, como lenguajes de *especificación*. Muchos DSLs están soportados por un compilador que genera aplicaciones a partir de programas escritos en el respectivo DSL. Bajo esta perspectiva, el compilador suele denominarse un *generador de aplicaciones* (Cleaveland, 1988) y el DSL suele denominarse un *lenguaje específico de la aplicación*. Existen, así mismo, otros DSLs no orientados a la producción o especificación de aplicaciones concretas, sino a la generación de bibliotecas o componentes (v.g. (Johnson, 1975) (Parr, 2007) (Crew, 1997) (Wang et al. 1997)). Otros DSLs están orientados a tareas de procesamiento específico de información, como la generación de, por ejemplo, presentaciones de documentos (v.g. (Knuth, 1984b)), dibujos (v.g. (Kernighan, 1981)), consultas a bases de datos (v.g. (Walmsley, 2007) (Beaulieu, 2005)), o transformación de documentos (v.g. (Tidwell, 2008)). En el ámbito de las aplicaciones de gestión, los DSLs suelen denominarse *lenguajes de cuarta generación* (4GL).
- Los DSLs están orientados a ser usados por los *expertos en el dominio* y/o por los *usuarios finales*. Efectivamente, el uso de DSLs está íntimamente relacionado con el concepto de *autoría*. De esta forma, la programación mediante DSLs puede identificarse como un tipo de *programación orientada al usuario*. Este tipo de programación surge cuando el usuario final

realiza tareas de programación sencillas, utilizando un lenguaje de macros o de *script*, quizá debidamente arropado por una interfaz de usuario apropiada. Un ejemplo típico es la programación de hojas de cálculo en sistemas como Excel (Young & Halvorson, 2003).

Por su parte, (Deursen et al., 2000) también identifica las siguientes ventajas aportadas por el uso de DSLs en el desarrollo de software:

- Los DSLs permiten la expresión de las soluciones en los términos y en el nivel de abstracción del dominio del problema. Consecuentemente, los propios expertos del dominio pueden entender, validar, modificar, e incluso desarrollar los programas en el DSL.
- Los programas en los DSLs son concisos, autodocumentados, y pueden ser reutilizados para diferentes propósitos. Esta ventaja es consecuencia directa del carácter declarativo de los mismos.
- Los DSLs aumentan la productividad, fiabilidad y mantenimiento (Kieburz et al., 1996) (Deursen & Klint, 1998), así como la portabilidad (Herndon & Berzins, 1988) de las aplicaciones.
- Los DSLs incorporan conocimiento de dominio, y así permiten la conservación y la reutilización de este conocimiento.
- Los DSLs permiten la validación y la optimización en el nivel del dominio (Basu et al., 1997) (Bruce, 1997) (Menon & Pingali, 1999).
- Los DSLs mejoran la prueba de las aplicaciones. Incluso, y tal y como se muestra en (Thibault, 1998) (Thibault et al. 1999), debido a su carácter específico, muchas veces será posible verificar automáticamente propiedades interesantes de la especificación que, en un lenguaje de propósito general, pueden resultar indecidibles.

Como contrapartida, (Deursen et al., 2000) identifica las siguientes desventajas en el uso de DSLs:

- Los costes de diseño, implementación y mantenimiento del DSL. El diseño del DSL puede involucrar actividades exhaustivas de análisis del dominio o de familias de programas. Por su parte, la implementación puede, en el peor de los casos, involucrar la construcción de un entorno de programación completo. Por último, en (Deursen & Klint, 1998) se analizan los problemas derivados del mantenimiento de los DSLs.
- Los costes derivados de la educación de los usuarios de los DSLs, que deben aprender el uso y las peculiaridades de dichos DSLs. El uso de criterios de diseño ortogonales y homogéneos en

características como sintaxis o interfaz de usuario de los DSLs finales puede ayudar a aminorar dichos costes.

- La disponibilidad limitada de los DSLs. Efectivamente, no siempre se dispondrá de un DSL para cada dominio o problema particular. Así mismo, y debido al carácter restringido de estos lenguajes puede ocurrir, como se ha indicado ya anteriormente, que no se disponga exactamente del DSL más apropiado para resolver el problema particular planteado.
- La dificultad de encontrar el alcance más adecuado para los DSLs. Efectivamente, la determinación de qué aspectos del dominio deben incluirse en un DSL y cuáles deben desdeñarse es un problema de diseño crucial en la configuración final del lenguaje.
- La dificultad de encontrar un equilibrio adecuado entre construcciones específicas del dominio y construcciones propias de un lenguaje de propósito general.
- La potencial pérdida de eficiencia en las aplicaciones producidas mediante un DSL, sobre todo cuando éstas se comparan con versiones equivalentes codificadas en un lenguaje de propósito general.

En (Thibault, 1998) (Deursen et al., 2000) (Mernik et al., 2005) se proporcionan multitud de dominios donde se han aplicado DSLs, así como ejemplos de dichos DSLs. En (Salus, 1998) se describe con detalle alguno de estos DSLs. Por su parte, en (Deursen, 1997) (Czarnecki & Eisenecker, 1999) (Kieburtz et al., 1996) se compara la aproximación basada en DSLs con otras aproximaciones a la generación de *software*, mientras que (Krueger, 1992) identifica el uso de DSLs como una aproximación central en reutilización de *software*, y compara dicho uso con otras aproximaciones a dicha reutilización.

2.3.3. Desarrollo de Software Dirigido por Modelos

La aplicación de modelos al desarrollo de software es una propuesta que, aunque tiene ya una larga tradición en Ingeniería del Software (véase, por ejemplo, (Krueger, 1992) (Neighbors, 1986)), ha experimentado un aumento espectacular de popularidad desde la creación del Lenguaje Unificado de Modelado (UML, del inglés *Unified Modeling Language*) (Booch et al., 2005) (OMG 2005) (OMG, 2005b). Esta propuesta se alinea completamente con los principios de desarrollo de software dirigido por lenguajes, y, en concreto, con los principios subyacentes a los lenguajes específicos de dominio. Efectivamente, el desarrollo de software dirigido por modelos propugna la definición de DSLs para cada dominio de aplicación que faciliten tanto la formulación de modelos en dichos dominios, como la posterior transformación de los mismos en el desarrollo de las aplicaciones finales (Stahl et al., 2006). Así mismo, dichos DSLs suelen ser de carácter gráfico, en contraposición al carácter textual propugnado por los enfoques analizados

anteriormente. Existen varias implementaciones y enfoques de desarrollo de software dirigido por modelos, entre los que destacan Model Driven Architecture (MDA) (Frankel, 2003), Agile Model-Driven Development (Ambler, 2003) (Ambler, 2004), Model Integrated Computing (Sztipanovits & Karsai, 1997) (Sprinkle, 2004), Domain-Oriented Programming (Thomas & Barry, 2003), Microsoft's Software Factories (Short et al., 2004) y Eclipse Modeling Framework (EMF) (Steinber et al., 2008).

En la formulación de DSLs, los enfoques dirigidos por modelos fomentan enfoques basados en metamodelado (Kleppe, 2008). De esta forma, la construcción de *metamodelos* (lo que habitualmente se denomina como *metamodelado*) es uno de los aspectos más importantes del desarrollo de software dirigido por modelos. Efectivamente, un metamodelo describe la posible estructura de modelos de la misma clase. Este aspecto es necesario para definir las *sintaxis abstractas* de los DSLs. En el contexto de los lenguajes informáticos, la sintaxis abstracta de un lenguaje caracteriza la representación interna de las frases de dichos lenguajes que es relevante de cara a llevar a cabo el procesamiento de dichas frases (Friedman et al., 2001). En lenguajes textuales, dicha sintaxis suele basarse en representaciones arborescentes. En el caso de los lenguajes gráficos habitualmente considerados en el desarrollo dirigido por modelos, dicha sintaxis se basa en grafos. En los apartados anteriores se han visto ejemplos de metamodelos, en relación con los EMLs, y también con los lenguajes de modelado de procesos de negocio. Los metamodelos se expresan mediante lenguajes y marcos de metamodelado. Dos ejemplos de dichos marcos son MOF (OMG, 2005c), piedra angular de la propuesta MDA, y el anteriormente citado EMF (Steinber et al., 2008).

Aparte del uso de metamodelos, el otro componente esencial de los enfoques dirigidos por modelos son las transformaciones entre modelos. Las transformaciones permiten caracterizar artefactos capaces de traducir automáticamente modelos que se ajustan a un determinado metamodelo origen, en modelos que se ajustan a un determinado metamodelo objetivo (transformaciones *modelo a modelo*). Así mismo, estas tecnologías permiten también realizar transformaciones entre modelos y otros formatos de representación (v.g. entre modelos y formatos textuales, dando lugar a las denominadas transformaciones de *modelo a texto* y de *texto a modelo*). Tal y como se describe en (Kleppe, 2008), en el desarrollo de DSLs, las transformaciones juegan un papel fundamental en:

- Dotar a los lenguajes de una o varias sintaxis concretas. Al contrario que la sintaxis abstracta, la sintaxis concreta de un lenguaje informático es la notación que el usuario final del lenguaje utiliza para escribir frases en dichos lenguajes. En desarrollo dirigido por modelos, es frecuente caracterizar esta sintaxis como un metamodelo que captura los elementos básicos de una notación (frecuentemente visual). De esta forma, para dotar al DSL de una sintaxis concreta

basta utilizar transformaciones que ligen dichos metamodelos orientados a notaciones con el metamodelo del lenguaje. Así mismo, utilizando este método, es posible equipar a un mismo DSL con múltiples sintaxis concretas.

- Dotar al DSL de una semántica operacional, mediante transformaciones que ligen sus metamodelos con metamodelos de menor nivel de abstracción. Por ejemplo, esta es una característica básica de MDA, donde se emplean transformaciones para convertir modelos independientes de plataforma (PIMs, del inglés *Platform Independent Models*) en modelos específicos de plataforma (PSMs, del inglés *Platform Specific Models*).

Por último, cabe destacar que los enfoques dirigidos por modelos orientados a usuarios finales propugnan la aparición de las anteriormente citadas plataformas de desarrollo dirigidas por lenguajes (Fowler, 2010). Este tipo de plataformas, mediante la integración consistente de metamodelos, transformaciones y técnicas generativas, facilitan sustancialmente la creación de editores especializados para cada DSL y, por tanto, la adopción de dichos DSLs por sus usuarios finales.

2.4. A modo de conclusión

Los EMLs permiten a profesores y educadores el diseño, la formalización y el intercambio de cursos basados en el concepto general de unidad de aprendizaje. Estas unidades de aprendizaje, debidamente formalizadas y descritas mediante metadatos (v.g. IEEE LOM) pueden re-almacenarse en una base de datos (también llamada repositorio) para su reutilización posterior, tanto para repetir el proceso de aprendizaje, como para la creación de unidades de aprendizaje nuevas tomando esas como punto de partida.

Además los EMLs tienen también otro uso, y es que sirven como medio de comunicación entre el personal técnico y el personal docente. El personal docente es responsable de la descripción de las experiencias educativas mientras que el personal técnico es responsable de la creación de intérpretes que permitan ejecutar automáticamente las UoLs creadas y su integración dentro del LMS que se utilice.

Al integrar el intérprete del EML dentro del LMS estamos proporcionando al personal docente la posibilidad de adaptar y personalizar el propio LMS, teniendo en cuenta sus necesidades educativas específicas, y sin demandar de ellos conocimientos de programación profundos. No obstante, a día de hoy, sí exige tener mucho conocimiento del lenguaje en sí debido a la ausencia de herramientas de autoría sencillas y maduras para usuarios finales. La aplicabilidad y explotación de los EMLs es una línea de investigación que actualmente se encuentra en desarrollo, ya sea centrada en la búsqueda de notaciones para documentar los diseños de aprendizaje, o en la

creación de lenguajes de diseño y ejecución de diseños de aprendizajes (Agostinho, 2009). Dentro de esta línea de investigación existe una tendencia novedosa basada en la creación de Lenguajes Visuales de Diseño Instruccional (VIDL, del término en inglés *Visual Instructional Design Languages*) que pueden ser considerados EMLs que tienen una notación visual asociada (Botturi et al., 2008) (Caeiro Rodriguez et al., 2010). Estos lenguajes tienen como objetivo facilitar la adopción de los EMLs simplificando la creación de los diseños de aprendizaje utilizando una notación gráfica visual.

La especificación IMS-LD se ha convertido en el estándar *de facto* como medio de incorporación en las herramientas de gestión de e-learning de aspectos más avanzados de metodologías educativas, lo que permite que los LMS trasciendan la concepción simplista de ser meros artefactos de distribución de contenidos educativos.

En la actualidad la comunidad investigadora considera que IMS-LD no está siendo aplicado más extensivamente debido en gran parte a la complejidad del propio lenguaje (Neumann et. al., 2009). No obstante, esta complejidad puede ser aprovechada, de modo que IMS-LD pueda ser utilizado como “lenguaje máquina” de modo similar al uso dado en esta Tesis (ver Capítulo 4) (Agostinho, 2009) (Karampiperis & Sampson, 2007) (Molina et. al., 2008). Por otro lado, otros autores proponen descartar la versión actual de IMS-LD y apostar por construir un nuevo EML con características similares a IMS-LD (pero más simple de utilizar) sobre especificaciones ya existentes y ampliamente utilizadas con el objetivo de promover y facilitar la adopción de una nueva especificación (Durand & Downes, 2009) (Durand et al., 2010).

Por tanto, IMS-LD está en un punto medio entre un modelo puramente educacional y un modelo puramente tecnológico. En nuestra opinión, para poder aplicar esta especificación de manera efectiva necesitamos movernos simultáneamente hacia ambos extremos. Efectivamente:

- Desde el punto de vista educativo, se necesita diseñar y documentar el proceso de enseñanza de manera cuidadosa, con el objetivo de que el diseño pueda ser analizado y/o reutilizado por otros docentes y que, finalmente, este diseño pueda ponerse en práctica (ejecutarse) automáticamente en un LMS. Por tanto es necesario elevar el nivel de abstracción proporcionado por las herramientas de autoría de UoL para que los docentes no tengan que conocer todos los detalles y complejidades del modelo subyacente impuesto por IMS-LD.
- Desde el punto de vista tecnológico, es necesario formalizar todos los detalles relativos a la ejecución de la UoL diseñada, por ejemplo, cuando se interpreta en un LMS. La especificación IMS-LD deja claramente fuera de su alcance los detalles específicos de ejecución de una UoL de modo que sería necesario un perfil de aplicación ampliamente aceptado que fijara dichos detalles y simplificara el desarrollo de entornos de ejecución, del mismo modo que SCORM ha

fijado detalles no incluidos en las especificaciones IMS en las que se basa para simplificar su implementación en sistemas comerciales.

Pensamos, así mismo, que la consecución práctica de estos objetivos puede lograrse adoptando los principios básicos de la Ingeniería de Lenguajes Software, y, principalmente, las modernas tendencias de desarrollo dirigido por modelos. Así mismo, pensamos que estas técnicas de desarrollo dirigido por lenguajes pueden complementarse también con la experiencia ganada con el uso de los lenguajes de modelado de procesos de negocio, tanto en lo referente a los aspectos operacionales (de ejecución de los modelos educativos), como a los aspectos de autoría de los diseños.

Capítulo 3

Objetivos y Planteamiento del Trabajo

Como se ha descrito en capítulos anteriores, las grandes expectativas creadas por la aparición de los Lenguajes de Modelado Educativo (EMLs) (Rawlings et al., 2002) (Koper, 2000) para lograr sistemas de e-learning mucho más versátiles y funcionales no se han visto cumplidas. Actualmente existen todavía distintos obstáculos que impiden adoptar de manera generalizada los EMLs y obtener los beneficios que, como se había enunciado previamente, son: sistemas más centrados en los alumnos, en los que las actividades educativas son los elementos clave y mediante los cuales se obtienen modelos educativos reutilizables e interoperables entre distintos sistemas de e-learning (Koper, 2001). Algunas de las causas que pueden explicar esta situación como, por ejemplo, la complejidad que el uso de los EMLs supone para una gran parte de los instructores, son problemas que consideramos hay que abordar desde el punto de vista metodológico (Burgos et al., 2007b). Otros problemas como, por ejemplo, la falta de herramientas de autoría para los EMLs, o la falta de reproductores en los sistemas e-learning que soporten las propuestas estándar de EMLs, son problemas que deben abordarse desde el punto de vista tecnológico (Griffiths et. al., 2005).

En este capítulo se comienza presentando los objetivos principales de esta Tesis. Seguidamente, se plantea la especificación de una solución metodológica y tecnológica que cubre estos objetivos. Esta especificación sirve como base para la propuesta de esta Tesis, que se denomina <e-LD> (Martínez-Ortiz et al., 2007b) y que se ha desarrollado como una plataforma para la autoría de diseños educativos con soporte a EMLs estandarizados como IMS-LD (IMS, 2003) (Koper & Tattersall, 2005).

3.1. Objetivos de la Tesis

El objetivo general de esta Tesis es proponer un enfoque basado en Ingeniería de Lenguajes Software que facilite la adopción y uso generalizado de los EMLs en los sistemas de *e-learning* y *b-learning* (aproximación combinada de educación presencial que hace uso de las TIC). Este

enfoque se fundamenta en el análisis crítico de los EMLs realizado en el capítulo anterior, donde se han identificado las principales características y limitaciones de este tipo de lenguajes. Este objetivo general se plasma en dos objetivos concretos:

- El primer objetivo es acercar los EMLs a los instructores. Puesto que los instructores son los usuarios finales de estos lenguajes, para facilitar su uso y promover su adopción es necesario que los EMLs se ajusten a la terminología empleada por dichos instructores y a la experiencia previa de los mismos.
- El segundo objetivo es promover la reutilización de diseños educativos existentes. La reutilización de diseños educativos previamente creados no suele ser directa ya que es necesario comprender el diseño y realizar un proceso de adaptación teniendo en cuenta las necesidades de los estudiantes a los que está dirigido el diseño. Por tanto, el proceso de reutilización debe recibir una atención equiparable al proceso de autoría de diseños educativos.

Las dos siguientes subsecciones profundizan en cada uno de estos aspectos, sentando las bases para el planteamiento del trabajo de Tesis, que se esboza en la siguiente sección.

3.1.1. Facilitar la adopción de los lenguajes de modelado educativo

Los EMLs han surgido como resultado de un proceso de análisis y abstracción sobre las descripciones de los diseños educativos y metodologías utilizadas por los instructores. Como resultado del análisis se identifican los conceptos básicos que utilizan los instructores en la descripción de los diseños educativos. Una vez identificados estos conceptos básicos se realiza una abstracción y formalización de modo que no sólo se puedan representar los diseños educativos, sino que además puedan ser automatizados por un sistema informático. Aunque el proceso de análisis y abstracción seguido en cada una de las distintas iniciativas descritas en el capítulo 2 aplica enfoques muy diversos, sin embargo, se pueden distinguir dos categorías principales:

- *Enfoques específicos.* Estos enfoques parten de un conjunto de requisitos y necesidades específicas, habitualmente con el objetivo de obtener una solución rápida y operativa que cubra las necesidades iniciales. Se caracterizan por un modelo de proceso basado en el desarrollo rápido de las herramientas que soportan el EML, guiado por las necesidades específicas de los actores involucrados. De este modo que el proceso de análisis y abstracción está dirigido por el desarrollo de dichas herramientas. Es más, en muchos casos, la descripción y formalización del EML vienen dadas por las herramientas en vez de realizarse de manera explícita e independiente. La principal ventaja de estos enfoques es la integración de los instructores dentro del proceso de desarrollo y, puesto que el desarrollo es específico, la posibilidad de adaptarse rápidamente a sus necesidades. Sin embargo, estos enfoques tienen como problema principal la dependencia de los desarrolladores de software (v.g. comunidad de software libre,

empresa privada) que desarrollan las herramientas de soporte de cada EML concreto. Como resultado, las herramientas y la expresividad del lenguaje pueden no adaptarse a las necesidades de los instructores que no hayan participado en el proceso de desarrollo. Así mismo, puede establecerse una dependencia con un sistema e-learning concreto a la hora de ejecutar los diseños educativos creados. Por último, pese a que no existe un modelo específico para la evolución del lenguaje, en el ámbito de comunidades de desarrollo activas, el proceso de evolución del lenguaje puede ser rápido.

- *Enfoques generales.* Estos enfoques parten del análisis de un amplio conjunto de escenarios educativos provenientes de distintas comunidades de instructores y tienen por objetivo crear un EML suficientemente expresivo como para permitir, como mínimo, describir los escenarios educativos de partida. Dentro de estos enfoques se encuentran las iniciativas de EMLs estándar que, además del objetivo anterior, tratan de garantizar la interoperabilidad entre distintos sistemas de los diseños producidos. Con este propósito, durante el proceso de definición del lenguaje se tienen en cuenta, no sólo las necesidades expresivas de los instructores, sino también otras consideraciones técnicas, e incluso requisitos adicionales, para lograr la compatibilidad con otras especificaciones estándar. Las ventajas más destacables de estos enfoques son: (i) la gran expresividad del EML resultante, que permite contemplar un mayor abanico de diseños educativos, (ii) la interoperabilidad de los diseños producidos y, (iii) la no dependencia de desarrolladores de software concretos, permitiendo tanto los desarrollos abiertos como los propietarios. No obstante, estos enfoques también presentan desventajas como, por ejemplo, la complejidad de uso de los EMLs resultantes o las limitaciones en el proceso de evolución de dichos EMLs. Estos EMLs son, en general, bastante complejos de utilizar debido principalmente a su alta expresividad, llegando a ser tan complejos como utilizar directamente un lenguaje de programación de propósito general (Beetham, 2004) (Heyer et. al., 2007) (Le Pallec et. al. 2006). Por otro lado, aunque durante el proceso de análisis se tienen en cuenta un gran conjunto de escenarios educativos, el EML resultante puede no satisfacer completamente las necesidades específicas de una comunidad de instructores. Sin embargo, la modificación y evolución del EML sigue necesariamente un proceso más lento debido a que se deben tener en cuenta no sólo las nuevas necesidades, sino también otros factores, como la compatibilidad con el EML actual o los requisitos de los desarrolladores de las herramientas.

Un ejemplo de solución desarrollada conforme con un enfoque específico es la aportada por la plataforma LAMS (Dalziel, 2003) (Dalziel, 2006) cuyo desarrollo se centra en los dos actores principales del proceso educativo: instructores y estudiantes. En particular, LAMS proporciona una herramienta versátil y fácil de utilizar para crear diseños educativos (denominados *secuencias*

LAMS en este contexto) mediante el uso de una notación visual. No obstante, los diseños educativos basados en secuencias LAMS únicamente están soportados por la herramienta LAMS tanto desde el punto de vista de ejecución como desde el punto de vista de autoría¹.

El ejemplo más representativo de EML desarrollado siguiendo el modelo general es IMS Learning Design (IMS-LD) (IMS, 2003). IMS-LD padece los problemas descritos para los enfoques generales y, además, sufre de una falta de herramientas de soporte, particularmente en el aspecto de autoría, que los mitiguen. Estas circunstancias han provocado la paradoja de que, aunque IMS-LD tiene el mayor potencial como el EML estándar de facto, no haya sido ampliamente adoptado por la comunidad docente ni soportado en la mayoría de los LMS comúnmente utilizados. Se ha entrado en un círculo vicioso en el que IMS-LD no se adopta por la falta de herramientas de soporte fáciles de utilizar y no hay herramientas de IMS-LD porque no hay una comunidad suficientemente amplia que atraiga al sector privado o a las comunidades de software libre. Esta situación tiene como resultado que la especificación no evolucione de forma efectiva, eliminando los defectos e imprecisiones identificados durante su puesta en práctica, o incorporando nuevas características no consideradas en la especificación original. Consideramos que esta situación es debida, en gran parte, a que junto con la especificación no se ha proporcionado un modelo de referencia que implemente la especificación completamente, con el objetivo de suplir las deficiencias e inexactitudes de la especificación y sirviendo como prototipo para desarrollos de herramientas de soporte reales. Por ejemplo, la especificación ADL SCORM (ADL, 2009) proporciona una implementación de referencia lo que ha simplificado mucho el desarrollo y la adopción de esta especificación de e-learning.

Ambos tipos de enfoques, específicos y generales, ofrecen ventajas interesantes. Por un lado, los enfoques específicos reaccionan rápidamente a las necesidades de los instructores, e integran a los mismos en el desarrollo, evaluación y evolución de las herramientas. Por otro lado, los enfoques generales promueven la aparición de un mayor número de iniciativas independientes y diferenciadas que comparten un mismo marco de referencia, permitiendo que desarrollos heterogéneos puedan ser integrados y utilizados de manera colaborativa. Por tanto, los enfoques específicos atraen a los instructores debido a su facilidad de uso, y los enfoques generales atraen a los desarrolladores de software debido a que permiten que sus herramientas sean utilizadas por una comunidad más amplia.

En este trabajo consideramos que ambos enfoques son, en realidad, complementarios, lo que nos lleva a formular el primero de los dos objetivos concretos de esta Tesis:

¹ Mediante el desarrollo de módulos específicos estos diseños actualmente pueden ejecutarse en diversos sistemas de gestión de e-learning como Moodle o Blackboard.

Objetivo 1. Facilitar la adopción de los EMLs por parte de los instructores, integrándolos en el proceso de desarrollo de EMLs específicos que faciliten la creación de diseños educativos (Martínez-Ortiz et al., 2008b). Al mismo tiempo se debe mantener la compatibilidad con las propuestas estandarizadas de EMLs para reutilizar la experiencia, herramientas e integración con plataformas existentes compatibles con las propuestas estándar (Martínez-Ortiz et al., 2009).

3.1.2. Fomentar la reutilización de diseños educativos

Uno de los objetivos principales de los EMLs es posibilitar la interoperabilidad, es decir, facilitar el intercambio de los diseños educativos entre distintos sistemas de e-learning. Los EMLs cumplen este objetivo definiendo un formato de intercambio (habitualmente un vocabulario XML) que es fácilmente procesable por una aplicación software (Birbeck et al., 2001). No obstante, aunque la interoperabilidad está ligada a la reutilización de diseños educativos, nosotros consideramos que éste sólo es un primer paso, ya que si no se facilita el análisis y comprensión de los diseños preexistentes, aunque se logre la interoperabilidad técnica no se resuelve de forma efectiva el problema asociado de la reutilización. Por ejemplo, utilizando las herramientas actuales es muy complejo partir de un diseño existente y mejorarlo o particularizarlo para un nuevo escenario de aplicación, llegando a ser necesario ejecutar el diseño educativo en un reproductor para comprender realmente su comportamiento.

Las características de los EMLs existentes, en particular sus expresividades y sus concepciones no centradas en la usabilidad, dificultan la reutilización de los diseños educativos expresados con estos EMLs. Por otro lado, existe un amplio consenso respecto a que los instructores deben comprender un diseño educativo como paso previo a su reutilización (Hernández-Leo et. al., 2006). De esta forma:

- Si el EML tiene gran expresividad, el proceso de reutilización de un diseño educativo sólo es factible si el diseño es suficientemente simple, ya que dicha reutilización implica navegar por la herramienta de autoría o, en el peor de los casos, enfrentarse a la representación XML del diseño educativo para analizarlo y, finalmente, comprenderlo. En el caso de diseños más elaborados esto es tan complejo que el modo más directo para analizar y comprender un diseño educativo pasa por su ejecución y prueba.
- Adicionalmente, si el EML no ha sido concebido teniendo en cuenta la usabilidad del mismo desde el punto de vista de los instructores, el instructor necesitará un profundo conocimiento para distinguir entre las características relevantes desde el punto de vista educativo y las características relevantes desde el punto de vista tecnológico. Es necesario, por tanto, que durante la concepción de los EMLs se tenga en cuenta la usabilidad del lenguaje por parte de

los instructores. Además se deben proporcionar un conjunto de herramientas específicas que faciliten la comprensión de los diseños existentes y que incluyan mecanismos para poder analizarlos sin que el usuario tenga que ser un experto en las tecnologías subyacentes (v.g. XML y estándares educativos).

De hecho, nosotros creemos que es necesario considerar conjuntamente los procesos de reutilización y autoría de diseños educativos (Martínez-Ortiz et al., 2008b). Idealmente el proceso debería ser circular, de modo que una misma herramienta pudiera no sólo producir un diseño educativo, sino también facilitar la comprensión y modificación de diseños ya existentes. No obstante creemos que este último aspecto (ayudar a comprender y modificar diseños pre-existentes) no se cumple simplemente ofreciendo la opción de poder importar un diseño existente en la herramienta de autoría, sino que, en la línea del primer objetivo de este trabajo, es necesario representar el diseño educativo con una notación que sea más cercana a los conceptos que utiliza habitualmente el instructor. Estas consideraciones nos llevan a formular el segundo objetivo de este trabajo:

Objetivo 2. Proponer un enfoque *completo* del proceso de creación de diseños educativos, que contemple tanto la autoría como la reutilización, para fomentar y simplificar el uso de EMLs por parte de los instructores (Martínez-Ortiz et al., 2009c).

3.2. Planteamiento del Trabajo: La Propuesta <e-LD>

Partiendo de las consideraciones anteriores y de los objetivos propuestos, podemos resumir que el propósito de este trabajo es formular soluciones que faciliten la adopción de los lenguajes de modelado educativo a los instructores y que, al mismo tiempo, fomenten la reutilización de los diseños educativos existentes.

De los objetivos planteados en la sección 3.1 se derivan una serie de requisitos específicos, tanto desde el punto de vista metodológico como desde el punto de vista técnico, que son el punto de partida para enunciar el núcleo de este trabajo de Tesis, y que se concretan en una propuesta denominada <e-LD>.

Aunque tanto la discusión y objetivos de la sección 3.1 como los requisitos que se presentan en esta sección se pueden aplicar a otros EMLs (ver trabajo futuro en el Capítulo 5), en este trabajo se ha seleccionado IMS-LD como EML representativo para poner en práctica la propuesta <e-LD>. La razón para seleccionar IMS-LD como caso de estudio se debe a que, como se ha indicado anteriormente, este EML goza de un alto grado de estandarización, así como de una elevada capacidad expresiva. No obstante, como hemos destacado en secciones anteriores, continúan

existiendo serias dificultades para su aceptación por parte de la comunidad de instructores, así como para su inclusión generalizada en sistemas e-learning de uso habitual.

3.2.1. Facilitar la adopción de los lenguajes de modelado educativo

3.2.1.1. Integración de lenguajes de modelado educativo de autoría y lenguajes de modelado educativo de intercambio

Para lograr el doble objetivo de facilitar la adopción de los EMLs y, a la vez, mantener la compatibilidad con los estándares, nuestro enfoque particular es distinguir entre dos tipos de EMLs, EMLs *de autoría* y EMLs *de intercambio* (Martínez-Ortiz et al., 2008b), y plantear cómo es posible combinarlos, de modo que logremos, al menos parcialmente, las ventajas aportadas por ambos tipos de EMLs:

- Los EMLs de autoría son lenguajes desarrollados de acuerdo a los enfoques específicos a los que se ha hecho alusión en la sección anterior. De esta forma, estos EMLs son lenguajes específicos de dominio más cercanos a la notación y vocabulario que utilizan los instructores para describir sus diseños educativos. Estos EMLs se concentran en los conceptos clave desde el punto de vista educativo, obviando la mayor parte de los detalles técnicos necesarios para su procesamiento automático. Además, estos EMLs tienen una expresividad limitada y adaptada a las necesidades específicas de un instructor o de una comunidad concreta. Sin embargo, apoyándose en este enfoque basado en lenguajes, estos EMLs no se ven como entes estáticos y fijos ya que, si se considera necesario, se pueden hacer evolucionar para adecuarlos a las nuevas necesidades de expresividad demandadas por el instructor o por la comunidad de instructores.
- Por otro lado, los EMLs de intercambio son EMLs desarrollados de acuerdo a los enfoques generales. De esta forma, se corresponden con propuestas estándar de EMLs que tienen una gran expresividad para dar soporte a una gran cantidad de escenarios educativos y, por tanto, una mayor independencia de dominio. Además, en estos EMLs de intercambio es difícil abstraerse de algunas de las características tecnológicas de más bajo nivel, que son necesarias para simplificar la interoperabilidad entre herramientas y plataformas.

Para que esta aproximación integrada sea viable, es necesario disponer de un proceso de transformación, preferentemente sin la intervención del instructor, que permita traducir los diseños educativos creados con un EML de autoría a un EML de intercambio concreto.

En <e-LD> se ha propuesto un EML de autoría orientado a describir las actividades y su secuenciación (es decir, el orden en el que deben de llevarse a cabo dichas actividades) para maximizar la eficacia del proceso de aprendizaje. Como ya se ha mencionado, como EML de

intercambio se ha elegido IMS-LD, de modo que los diseños educativos finalmente son traducidos a una unidad de aprendizaje (UoL) de IMS-LD.

3.2.1.2. Modelo de proceso completo que integre a instructores y desarrolladores

Este enfoque integrador, que combina los dos tipos de EMLs, debe incluir un modelo de proceso completo que, por un lado, contemple las distintas fases de provisión, uso y evolución del EML de autoría y que, por otro lado, determine el papel a realizar por los principales actores (los instructores y los desarrolladores) en cada una de dichas fases:

- En la primera fase de provisión del EML de autoría, los desarrolladores son los encargados de formalizar los conceptos clave, que previamente han sido identificados y facilitados por los instructores, y que son necesarios para describir los escenarios educativos contemplados. Los EMLs de autoría sirven de lengua franca entre instructores y desarrolladores. De este modo, cada EML de autoría no sólo sirve a las necesidades del instructor, sino que facilita la comunicación entre estas dos comunidades tan dispares.
- Una vez completada esta fase de provisión del EML de autoría específico, se pasa a la fase de uso del EML, en la que los instructores utilizan el EML de autoría producido para describir los escenarios y estrategias educativas que desean formalizar.
- Finalmente, la fase de evolución del EML es la actividad que permite evolucionar el EML de autoría a partir de las nuevas necesidades expresivas planteadas por el instructor y formalizadas por los desarrolladores.

Las técnicas de ingeniería de lenguajes en general, y de los lenguajes específicos de dominio en particular, permiten establecer este modelo de proceso integrador, centrado en las necesidades de los instructores y que, al mismo tiempo, facilita la comunicación y colaboración entre desarrolladores e instructores.

3.2.1.3. Propuesta de una notación gráfica para el EML de autoría

Las notaciones gráficas se utilizan con éxito en distintos dominios y, en muchos casos, con varios propósitos. Por ejemplo, una notación gráfica puede servir, por un lado, como herramienta de diseño, y, por otro lado, como herramienta de documentación o análisis. Un caso particular de esta situación es la notación gráfica que incluye el Lenguaje de Modelado Unificado (UML) (OMG 2005) (OMG, 2005b) (Booch et al., 2005) y que se utiliza ampliamente en el dominio de la Ingeniería del Software para modelar sistemas complejos. Los diagramas creados con esta notación durante las fases de diseño y conceptualización habitualmente pasan a formar parte de la documentación final del sistema, siendo especialmente útiles para desarrolladores que se incorporen posteriormente al proyecto o quieran reutilizar parte o la totalidad del sistema software.

Siguiendo este mismo enfoque, como parte de nuestra propuesta <e-LD> se ha definido una notación gráfica para una herramienta de autoría denominada <e-LD> Author. En base a experiencias previas, como por ejemplo, la ya citada para LAMS (Dalziel, 2006), existe evidencia de que el uso de la notación gráfica facilita la creación de nuevos diseños educativos complejos y reduce la carga cognitiva del instructor. Además, los diagramas creados con la notación gráfica son de mayor utilidad durante el proceso de reutilización de un diseño educativo (Hernández-Leo et al., 2006). Con la notación gráfica, el instructor que intenta reutilizar el diseño (es decir, un instructor que no ha sido el creador del mismo) obtiene una vista rápida general de la estructura de dicho diseño, simplificando, por tanto el análisis y la comprensión del mismo.

La aplicación de la ingeniería de lenguajes posibilita la creación de múltiples notaciones para el mismo lenguaje (Gronback, 2009) (Kleppe, 2008). De este modo, el lenguaje se formaliza mediante un *modelo abstracto* que define la estructura del lenguaje y sobre el cual se definen una o varias *notaciones concretas* que permiten crear instancias concretas del modelo abstracto. Esta separación posibilita que en las instancias del modelo abstracto se pueda trabajar, por ejemplo, con distintas notaciones gráficas para adaptarlas a las necesidades específicas de los instructores (en la misma línea que se hace con los DSLs).

En <e-LD> se han propuesto dos notaciones concretas principales: una notación gráfica para la autoría y una notación para almacenar los modelos. La notación gráfica es la notación que usan los instructores para crear diseños educativos. Aunque actualmente esta notación está inspirada en UML e incluye conceptos de diagramas de flujo, es necesario destacar que esta notación puede adaptarse a las necesidades y preferencias de los instructores. Como notación para almacenar los modelos se utiliza el formato de serialización nativo de Eclipse EMF (Steinber et al., 2008), formato muy similar al formato XML Metadata Interchange (XMI) utilizado para almacenar e intercambiar modelos siguiendo el estándar Meta Object Facility (MOF), ya que Eclipse EMF ha sido utilizado como *framework* para describir la sintaxis abstracta del lenguaje de representación de diseños educativos en <e-LD>.

3.2.2. Fomentar la reutilización de diseños educativos

3.2.2.1. Enfoque completo al proceso de autoría y reutilización de diseños educativos

Como se ha discutido previamente, en el caso particular de IMS-LD la expresividad del lenguaje dificulta la comprensión y reutilización de los diseños educativos. Es más, esta riqueza expresiva de IMS-LD hace que, desde un punto de vista teórico, no sea factible la traducción automática a la notación gráfica propuesta en <e-LD>. Por tanto, será necesario considerar únicamente técnicas semiautomáticas. Incluso si el proceso de traducción se restringiera a subconjuntos tratables de IMS-LD, éste debería abordar el problema añadido de que no sólo basta

con conseguir una representación con la misma semántica en otro lenguaje, sino que además la representación resultante debe ser comprensible para los instructores. Por ejemplo, en un proceso cíclico, en el que primero se representara el modelo educativo en la notación gráfica <e-LD> y, posteriormente se exportara éste a la representación de XML de IMS-LD, la reimportación en <e-LD> no tendría porque producir exactamente el mismo diseño, sino únicamente uno semánticamente equivalente, si bien no es equivalente desde el punto de vista, siempre subjetivo, de la comprensibilidad.

En <e-LD> se plantea un enfoque completo al proceso de autoría que incluye los distintos aspectos: (i) la propuesta de una notación gráfica de alto nivel, (ii) el planteamiento de una metodología de diseño educativo orientado a la definición del flujo (secuenciamiento) de actividades, (iii) la compatibilidad con estándares de e-learning mediante exportación automática y, (iv) el proceso semiautomático de importación desde representaciones XML de diseños educativos IMS-LD. En este último aspecto del proceso de importación semiautomática se identifica como necesaria la colaboración de un experto en la tecnología (tanto en XML como en el estándar IMS-LD) que ayude a representar en <e-LD> aquellos conceptos más complejos que actualmente no pueden ser importados de forma automática.

3.2.2.2. Herramientas de análisis de diseños educativos para facilitar su reutilización

Las herramientas de autoría están específicamente diseñadas para simplificar el proceso de autoría y, habitualmente, facilitan también la reutilización de diseños previos realizados con la misma herramienta y notación. Sin embargo, el soporte para simplificar la reutilización de un diseño educativo existente en otra notación es un proceso mucho más difícil. En el apartado anterior hemos abordado brevemente el proceso de importación semiautomática con ayuda de un experto, pero realmente hemos identificado otra actividad clave y previa a la propia importación para lograr una reutilización efectiva, que es ayudar a entender realmente que “hace” o que “representa” un diseño educativo preexistente. La representación XML de los diseños IMS-LD es demasiado compleja para que, sin ser un experto tanto en la tecnología XML como en la propia especificación IMS-LD, se puedan entender la mayor parte de dichos diseños y, por tanto, se puedan intentar reutilizar o emplear como base para nuevos diseños.

La propuesta de este trabajo es proporcionar herramientas que ayuden a dicha comprensión en alguno de los aspectos que se han identificado como más complejos de entender, como son los elementos que intervienen en la representación del diseño y las complejas relaciones que se establecen entre ellos. Con este propósito se plantean dos herramientas de análisis:

- Un navegador de UoL. Esta herramienta facilita una vista hipertextual que permite navegar a través de los distintos elementos que constituyen el diseño educativo. Su misión es facilitar la comprensión estática o estructural del diseño. Este navegador proporciona dos vistas: una

resumida y otra detallada. La vista resumida proporciona una lista de elementos que componen el diseño y, para cada uno de ellos, ofrece una sinopsis del elemento, de modo que sirve principalmente para familiarizarse con los distintos componentes que intervienen. La vista detallada proporciona un acceso sencillo al extracto del documento XML relativo a cada uno de los elementos del diseño, y sirve para obtener información más concreta y detalles específicos (v.g. propiedades, reglas).

- Un visualizador de dependencias. Esta herramienta permite visualizar de forma gráfica las distintas interdependencias existentes entre los elementos del diseño educativo. Esta herramienta permite analizar y anticipar en tiempo de diseño el comportamiento que tendrá el diseño educativo en tiempo de ejecución. Por tanto, esta herramienta simplifica un proceso de análisis que, de otra manera, sería especialmente complejo, al implicar la exploración directa de la representación XML del diseño. Tal complejidad se debe al potente sistema de reglas que incluye IMS-LD y a que las relaciones entre elementos viene determinada por una combinación de reglas y propiedades que puede provocar gran cantidad de efectos laterales.

3.3. A modo de conclusión

Con el objetivo de promover la adopción y aplicación de los lenguajes de modelado educativo, en esta Tesis planteamos dos objetivos principales: facilitar la aplicación de los EMLs por parte de los instructores y facilitar la reutilización de diseños educativos existentes. Cubrir estos objetivos contribuirá a reducir algunos de los principales problemas que impiden la adopción de los EMLs en general y de IMS-LD en particular.

A fin de demostrar que, en la práctica, es posible satisfacer estos objetivos, se plantea como trabajo de esta Tesis la propuesta <e-LD>, que abordará de manera metodológica y técnica los requisitos establecidos en la sección 3.2.

El siguiente capítulo aporta una discusión integradora del contenido de los artículos que dan soporte a la presente Tesis. Esta discusión relaciona los contenidos de dichas publicaciones con la declaración de objetivos presentada en este capítulo. Por otro lado, el Capítulo 5 incluye una discusión en la que se analiza el grado de cumplimiento de estos objetivos así como las principales líneas de trabajo futuro.

Capítulo 4

Discusión de las contribuciones de los artículos

Para lograr los objetivos planteados y desarrollar la propuesta descrita en el capítulo anterior se han seguido las cuatro etapas siguientes:

- Se ha comenzado analizando las analogías existentes entre el proceso de diseño educativo basado en EMLs y los enfoques de desarrollo basados en lenguajes específicos de dominio. El propósito de esta etapa es mostrar cómo los EMLs pueden entenderse, en realidad, como lenguajes específicos del dominio educativo. Efectivamente, tal y como se ha indicado en el Capítulo 2 el desarrollo de software basado en lenguajes específicos de dominio es un campo sólidamente cimentado y con una fuerte tradición, en el que existen métodos sistemáticos, técnicas y herramientas bien definidas. Esta madurez simplifica el proceso de concepción, diseño e implementación de nuevos lenguajes específicos orientados a cubrir las necesidades de los expertos en un determinado dominio. De esta forma, la caracterización de los EMLs como un caso particular de DSLs permite extrapolar a los escenarios de creación de herramientas de soporte para EMLs los principios básicos utilizados en el desarrollo de DSLs (en particular, los modernos principios de desarrollo de DSLs dirigido por modelos).
- Una vez enmarcados los EMLs en el contexto general del desarrollo basado en DSLs, en una segunda etapa se ha analizado cómo aplicar sistemáticamente los principios básicos de la Ingeniería de Lenguajes Software para facilitar el desarrollo de distintos tipos de herramientas de soporte para EMLs, con el fin de promover el uso de este tipo de lenguajes en el entorno educativo.
- La aplicación de los principios de Ingeniería de Lenguajes Software al desarrollo de herramientas para EMLs promueve la caracterización de una arquitectura integradora de las

facetas de autoría y operacionalización de diseños educativos basados en EMLs formales. De esta forma, en una tercera etapa se ha formulado una arquitectura genérica para la autoría y operacionalización de diseños educativos expresados mediante EMLs que sirve como marco para la incorporación sistemática y coherente de las herramientas de soporte desarrolladas de acuerdo a los principios de la Ingeniería de Lenguajes Software.

- Finalmente, en la última etapa del desarrollo de esta Tesis se han aplicado los resultados obtenidos en las etapas anteriores en el desarrollo de un caso de estudio que muestra cómo es posible consumir los objetivos propuestos en el capítulo anterior.

Este capítulo describe las publicaciones editadas en las que se presentan los principales resultados de investigación obtenidos en cada una de estas etapas. De esta forma, estos artículos describen los diferentes aspectos relativos a: la identificación de los EMLs como un tipo particular de lenguajes específicos de dominio (sección 4.1), la aplicación de las técnicas de Ingeniería de Lenguajes Software en el desarrollo de aplicaciones educativas basadas en EMLs (sección 4.2), la propuesta general <e-LD> orientada tanto a la autoría como a la instrumentación de EMLs (sección 4.3), y la forma en la que esta propuesta aborda los objetivos específicos de investigación propuestos en esta Tesis (sección 4.4).

4.1. Lenguajes de Modelado Educativo como Lenguajes Específicos de Dominio

4.1.1. Análisis del Dominio de los Lenguajes de Modelado Educativo

En (Martínez-Ortiz et al. 2007) se realiza un análisis de las principales iniciativas de EMLs existentes en el dominio de la Informática Educativa. Como resultado de dicho análisis, se identifican dichos lenguajes como un tipo particular de lenguajes específicos de dominio. Efectivamente:

- Los EMLs son lenguajes que permiten configurar y adaptar de manera totalmente flexible las herramientas educativas en términos no tecnológicos, sino didácticos. Desde este punto de vista, los EMLs permiten simplificar las tareas de gestión de herramientas que actualmente utiliza un profesor en entornos de e-learning y b-learning. Los EMLs permiten definir formalmente la descripción y organización de las actividades que forman parte de un diseño educativo. Como resultado, esta descripción puede ser procesada mediante una aplicación software, de modo que se puede utilizar para configurar el LMS de la organización según las necesidades descritas en un diseño educativo totalmente formalizado. De esta forma, los usuarios de los EMLs no son tecnólogos, sino expertos en el dominio educativo: profesores, diseñadores instruccionales, etc.

Estos permiten mantener una cierta independencia entre profesores y tecnólogos para poder llevar a cabo tareas más complejas que habitualmente se pueden realizar a través de las interfaces de usuario de los LMSs. En realidad se puede decir que, al menos hasta cierto punto, devuelven el control de la personalización de dichas plataformas a los expertos en Educación de acuerdo a sus necesidades y preferencias particulares.

- Los EMLs sirven de lenguaje de comunicación común entre tecnólogos y expertos en Educación. Efectivamente, los EMLs establecen un vocabulario común comprensible tanto por desarrolladores como por profesores. Como resultado, se mitigan los problemas de comunicación a los que habitualmente se enfrentan los miembros de los equipos multidisciplinares típicos en el desarrollo de aplicaciones educativas, y que se derivan de formaciones, campos de experiencia y jergas muy heterogéneos, ya que dichos miembros provienen de dominios muy diversos (v.g. informáticos, pedagogos).

En este trabajo también se identifica la necesidad de mejorar la usabilidad de las herramientas de soporte para los EMLs, manteniendo, al mismo tiempo, sus características como lenguajes abiertos y estandarizados. Este aspecto de la estandarización es relevante para que las soluciones aportadas puedan usarse de forma general en distintos sistemas y plataformas.

4.1.2. Gestión de Contenidos Mediante Lenguajes de Mercado Descriptivos Específicos de Dominio

En (Martínez-Ortiz et al., 2006c) se explora un enfoque *top-down* al proceso de provisión de EMLs. Para ello se parte de un DSL ya existente, que se especializa para facilitar la creación, gestión y publicación de contenidos educativos. El lenguaje elegido en este trabajo es DocBook (Walsh & Muellner, 1999), un lenguaje XML orientado principalmente a la creación de manuales en dominios técnicos y en formato *libro*, que se especializa para potenciar los procesos avanzados de producción de contenidos educativos. De esta forma, el sistema resultante, que se denomina <e-DocBook>:

- Aprovecha directamente toda la maquinaria de publicación de DocBook para soportar un modelo de producción basado en una fuente única (archivos XML) y múltiples formatos destino (PDF, HTML, etc.)
- Especializa el lenguaje para permitir producción avanzada de contenidos educativos.

Más concretamente, <e-DocBook>:

- Extiende DocBook con marcado que permite la generación de diapositivas, y enriquece el repertorio de hojas de estilo XSL asociadas al lenguaje para permitir la generación automática de dichas diapositivas a partir de los propios documentos con los materiales educativos.

- Incorpora, así mismo, una utilidad que, en base a la estructura del documento en capítulos, apartados, secciones, etc., fragmenta el mismo en múltiples subdocumentos separados, genera un manifiesto de IMS Content Packaging que refleja, en su organización dicha estructura y empaqueta todo ello de forma estándar en un paquete IMS desplegable en cualquier plataforma educativa que soporte dicho formato de intercambio (v.g. Moodle, Blackboard, etc.). Así mismo, extiende DocBook con marcado que permite representar los metadatos asociados con los diferentes niveles de organización.
- Extiende DocBook con marcado que permite adaptar los contenidos en base a diferentes niveles de lectura y diferente granularidad en la presentación de los contenidos finales.

4.2. Ingeniería de Lenguajes Software y Lenguajes de Modelado Educativo

En (Martínez-Ortiz et. al., 2009b) se explora cómo aplicar de forma sistemática técnicas de Ingeniería de Lenguajes Software en el diseño y operacionalización de EMLs más avanzados. Para ello se toma como caso de estudio un EML de secuenciamiento de actividades que abstrae las principales características expresivas de la propuesta <e-LD> realizada en esta Tesis. De esta forma, en este trabajo se describe cómo aplicar técnicas de Ingeniería de Lenguajes Software para:

- Llevar a cabo el diseño del núcleo del EML. Para ello, se propone: (i) la caracterización de la sintaxis abstracta del lenguaje mediante un metamodelo adecuado, (ii) la serialización de dicho metamodelo mediante una notación textual basada en conjuntos de predicados (al estilo de los hechos básicos en un programa Prolog), y (iii) la expresión formal de la semántica operacional del lenguaje mediante un conjunto de reglas de inferencia, siguiendo el estilo de la especificación estructural de semánticas operacionales de lenguajes de programación (Mosses, 2004) (Mosses, 2006) (Plotkin, 2004)).
- Utilizar dicho núcleo para soportar todo un cúmulo de desarrollos posteriores focalizados a validar el diseño y proporcionar herramientas de soporte para el lenguaje.

De esta forma, el trabajo muestra cómo, una vez definido el núcleo lingüístico básico, es posible:

- Producir prototipos rápidos del lenguaje. Para ello, el artículo propone utilizar el lenguaje Prolog para: (i) codificar la semántica operacional directamente como cláusulas Prolog, (ii) utilizar un mecanismo basado en flujos (*streams*) y el mecanismo de corrutinas en Prolog, para separar los aspectos relativos a la ejecución de los diseños de los aspectos relativos a la interacción con los diferentes participantes en el proceso educativo.

- Proporcionar sintaxis concretas para el EML, tanto visuales como textuales. Dichas notaciones están orientadas a facilitar tanto la autoría como el intercambio entre distintas herramientas de los diseños educativos expresados en el EML. El trabajo ejemplifica este aspecto con tres ejemplos: (i) notaciones visuales basadas en la notación de los diagramas de actividades UML, (ii) sintaxis embebidas en Prolog para facilitar la participación de los instructores en el proceso de prototipado rápido (se utiliza, para ello, las características de definición de nuevos operadores de dicho lenguaje) y (iii) lenguajes de marcado XML de propósito específico, con el objetivo de facilitar el intercambio de diseños. Así mismo, analiza la posibilidad de utilizar técnicas clásicas de desarrollo de procesadores de lenguaje, basadas en gramáticas de atributos, para soportar sintaxis textuales arbitrarias.
- Proporcionar herramientas específicas para la verificación de propiedades sobre los diseños educativos creados con el EML (v.g. garantía de finalización del proceso educativo). La idea básica es abstraer la semántica operacional para permitir traducir directamente los diseños en máquinas de estados, y utilizar algoritmos y herramientas de *model-checking* para llevar a cabo la prueba de las propiedades. Desde el punto de vista educativo estas técnicas pueden proporcionar algunas comprobaciones automáticas sobre los diseños creados para ayudar a evitar errores.
- Llevar a cabo la exportación a EMLs estandarizados (en concreto, a IMS-LD), así como la importación de diseños expresados en dichos EMLs. En particular, el artículo muestra cómo, en general, ambas actividades pueden automatizarse parcial o totalmente, dependiendo de las características estructurales y semánticas de los lenguajes involucrados.
- Llevar a cabo la construcción de entornos de ejecución de los diseños que preservan la especificación semántica de alto nivel de los núcleos de los EMLs.

4.3. El Enfoque <e-LD>

En (Martínez-Ortiz et al., 2007b) se presenta el enfoque genérico seguido en la propuesta <e-LD> de soporte a EMLs, tanto desde el punto de vista de la autoría de diseños educativos como desde el punto de vista de su operacionalización. Más concretamente:

- Desde el punto de vista de la autoría, se propone el uso de sintaxis visuales orientadas a flujos de actividades, así como el complemento de dichas sintaxis con elementos textuales. Se describe, así mismo, un primer piloto de este concepto, en el cuál se utilizan los diagramas de actividades UML para especificar la estructura de alto nivel del diseño, así como los aspectos relativos al secuenciamiento de las actividades. Dichos diagramas se crean con herramientas de edición de diagramas UML convencionales. Por su parte, los aspectos no expresables mediante

la notación gráfica se detallan textualmente, utilizando la herramienta Reload LD (Griffiths et al., 2008).

- Desde el punto de vista de la operacionalización, se propone la traducción automática de los diseños al lenguaje soportado por un motor de ejecución de flujos de procesos de negocio estándar. En particular, se realiza esta propuesta adoptando WS-BPEL (Curbera et al., 2002) (Andrews et al. 2003) (Barreto et al., 2007) como motor de ejecución de dichos flujos.

El trabajo muestra también cómo la arquitectura de <e-LD> permite de manera natural la incorporación de múltiples EMLs, así como la transformación automática o semi-automática entre diseños expresados en dichos EMLs. El trabajo también evidencia cómo el aspecto esencial de esta propuesta reside en disponer de un soporte adecuado que facilite la autoría de los diseños.

4.4. Autoría y Reutilización de Unidades de Aprendizaje en <e-LD>

En (Martínez-Ortiz et al., 2008b) se introduce la clasificación de los EMLs en las categorías de EMLs de intercambio y EMLs de autoría. De esta forma, se indica cómo, por una parte, los EMLs de intercambio están ideados para facilitar el intercambio de diseños educativos a nivel técnico entre diferentes herramientas y, por otro lado, los EMLs de autoría están ideados como EMLs de propósito específico especialmente creados para facilitar el proceso de autoría por parte de los instructores. De esta forma, los EMLs de autoría incluyen vocabularios y notaciones más cercanas a los instructores y, por tanto, a priori también más fáciles de usar. Adicionalmente, se propone además un modelo de proceso de trabajo colaborativo entre los instructores y el personal técnico para diseñar estos EMLs de propósito específico. Este modelo descansa sobre los fundamentos de la Ingeniería de Lenguajes Software dirigida por modelos, tanto para el diseño del EML de autoría y sus notaciones, como para la creación de las herramientas de soporte.

En (Martínez-Ortiz et al., 2009c) se detalla cómo este modelo genérico se particulariza de acuerdo con el enfoque <e-LD>, dando lugar a una herramienta de autoría denominada <e-LD> Author. En esta herramienta, el EML de intercambio es IMS-LD. De esta forma, tras realizar un análisis de los elementos estructurales y expresivos de IMS-LD, se concluye que IMS-LD (sobre todo en lo que se refiere al nivel B y a la autoría de diseños adaptativos) no es directamente apropiado como EML de autoría, sino como EML de intercambio. Por tanto, se diseña un EML de autoría en el que el secuenciamiento de actividades se realiza siguiendo un paradigma orientado a definir el flujo de actividades, en la línea de los diagramas de actividades UML anteriormente citados. Este enfoque es similar al utilizado en los lenguajes de modelado de procesos de negocio.

El otro aspecto clave abordado en (Martínez-Ortiz et al., 2009c) es tratar no únicamente los aspectos de autoría, sino también los aspectos de reutilización de diseños previamente existentes. Para ello, se propone un proceso genérico colaborativo para reutilizar un diseño de aprendizaje existente que esté formalizado en un EML de intercambio, y se aplica dicho proceso al caso particular de IMS-LD como EML de intercambio, y la notación orientada a flujo de <e-LD> Author como EML de autoría. En este caso particular, el proceso de reutilización es un proceso realizado de manera semiautomática debido a las diferencias sintácticas y semánticas entre el EML de autoría de <e-LD> Author e IMS-LD.

En (Martínez-Ortiz et al., 2009d) se detalla las herramientas de soporte al proceso semi-automático de reutilización, herramientas que se basan en las técnicas de ayuda a la comprensión de artefactos software utilizadas en Ingeniería del Software. El objetivo de dichas herramientas es facilitar el análisis de diseños de aprendizajes creados con IMS-LD, a través de la visualización explícita de las interdependencias entre los distintos elementos de un diseño educativo expresado en IMS-LD (actividades, reglas, propiedades, etc.). Estas herramientas de análisis son útiles no sólo en el contexto de <e-LD> Author, sino también como apoyo complementario para las herramientas de autoría específicas para IMS-LD que existen en la actualidad.

Por último, la transformación del EML de autoría de <e-LD> a IMS-LD sí es completamente automatizable. En (Martínez-Ortiz et al., 2009) se describe las bases para dicha traducción, mostrando cómo traducir un lenguaje de secuenciamiento de actividades orientado a flujo en un lenguaje basado en reglas. En este trabajo se deja constancia, así mismo, de la posibilidad de tener múltiples notaciones para el mismo EML, esbozándose una notación textual equivalente a la notación gráfica propuesta en <e-LD> Author.

Capítulo 5

Conclusiones y Trabajo Futuro

Los capítulos anteriores han contextualizado el dominio de los EMLs bajo la óptica de los lenguajes específicos de dominio y la Ingeniería de los Lenguajes Software, y se ha discutido cómo, tomando dicha contextualización como punto de partida, es posible construir herramientas que faciliten el uso de estos lenguajes a fin de simplificar tanto la creación de nuevos diseños educativos como la reutilización de diseños educativos ya existentes. Tal y como se ha discutido en el capítulo anterior, estos aspectos se detallan en las distintas publicaciones que integran esta memoria de Tesis. De esta forma, con el presente capítulo se concluye esta memoria, resumiendo las principales aportaciones (sección 5.1), y describiendo algunas líneas de investigación futuras (sección 5.2).

5.1. Principales Aportaciones

En esta sección se enuncian las principales aportaciones realizadas en esta Tesis. Más concretamente, y en base a la discusión mantenida en los capítulos precedentes, pueden destacarse las siguientes:

- Contextualización y análisis crítico de los Lenguajes de Modelado Educativo en el marco general de los Lenguajes Específicos de Dominio.
- Aplicación de la Ingeniería de los Lenguajes Software al desarrollo sistemático de herramientas de soporte de Lenguajes de Modelado Educativo.
- Propuesta de una arquitectura flexible para sistemas e-learning dirigidos por Lenguajes de Modelado Educativo.
- Particularización de la arquitectura en una herramienta, denominada <e-LD> Author, para simplificar la autoría y la reutilización de diseños educativos en IMS-LD.

Los siguientes puntos presentan con más detalle cada una de estas aportaciones.

5.1.1. Contextualización de los Lenguajes de Modelado Educativo en el marco general de los Lenguajes Específicos de Dominio

Una de las aportaciones de esta Tesis ha consistido en evidenciar las relaciones existentes entre el uso de EMLs en e-learning y el desarrollo de software basado en lenguajes específicos de dominio. Para ello, tras llevar a cabo un amplio análisis de diferentes lenguajes de modelado educativo utilizados en e-learning, se ha observado que:

- Por una parte, dichos lenguajes están orientados a describir los distintos aspectos del proceso educativo soportado por las plataformas, sistemas y aplicaciones e-learning. De esta forma, al incluir estructuras conceptuales cercanas a los expertos en educación, estos lenguajes pueden ser entendidos e incluso utilizados directamente por dichos expertos.
- Por otra parte, el carácter formal de dichos lenguajes permite su procesamiento automático mediante herramientas software. Además, en los EMLs de mayor capacidad expresiva, realmente posibilitan la configuración automática de las citadas plataformas, sistemas y aplicaciones para adaptarlas a las necesidades de los distintos agentes involucrados en el proceso de aprendizaje.

De esta forma, desde la óptica de los DSLs, los EMLs pueden considerarse como lenguajes de *script* muy específicos y orientados a los instructores expertos en el dominio, en lugar de a los desarrolladores informáticos. Por su parte, el papel de los desarrolladores no es ya el desarrollo de aplicaciones específicas, sino el diseño e implementación de lenguajes específicos y de las herramientas de soporte para estos lenguajes. Por tanto, desde esta óptica, la introducción de EMLs en el dominio de e-learning entronca directamente con la filosofía de desarrollo de software dirigido por lenguajes específicos de dominio. Como caso de estudio, en esta Tesis se ha llevado a cabo el desarrollo de <e-DocBook>, un sistema para la producción documental de contenidos educativos estandarizados basado en DocBook.

5.1.2. Aplicación de la Ingeniería de los Lenguajes Software al desarrollo sistemático de herramientas de soporte de Lenguajes de Modelado Educativo

La segunda aportación de esta Tesis ha sido mostrar cómo es posible utilizar los principios de la Ingeniería de los Lenguajes Software para normar el diseño sistemático de EMLs, así como el desarrollo sistemático de las herramientas de soporte para dichos lenguajes. Para ello se ha mostrado cómo el diseño de un EML descansa en dos pilares básicos:

- Por una parte, una caracterización estructural apropiada del lenguaje. Dicha caracterización puede tomar diferentes formas y ser denominada de muy diversos modos. Por ejemplo, en la

literatura clásica sobre lenguajes informáticos, dicha caracterización recibe el nombre de *sintaxis abstracta*, término también heredado por las modernas corrientes de Ingeniería de Lenguajes Software dirigida por modelos para hacer referencia a los metamodelos que caracterizan los lenguajes de modelado. Por su parte, en los esfuerzos de estandarización de IMS se suele hablar de *modelos de información*. En cualquier caso, y aplicados al campo de los EMLs, el fin último de todos estos esfuerzos es proporcionar una caracterización de la estructura abstracta de los diseños educativos, es decir, de los elementos básicos que han de tenerse en cuenta para hacer posible la posterior manipulación de dichos diseños en el interior de las herramientas.

- Por otra parte, una caracterización adecuada del comportamiento en tiempo de ejecución de los diseños. Esta caracterización es la *semántica operacional* del lenguaje. Este concepto ha sido estudiado con detalle en los enfoques clásicos al diseño e implementación de lenguajes informáticos, y ha sido adoptado también en los enfoques modernos a través del concepto de *transformación* entre modelos. Aplicado al campo de los EMLs, este concepto supone caracterizar la forma de orquestar la ejecución de estos diseños en el interior de las aplicaciones educativas.

En esta Tesis se ha evidenciado la factibilidad de combinar de manera adecuada los métodos clásicos con las tendencias modernas, promoviendo, por ejemplo, la caracterización de la sintaxis abstracta de los EMLs mediante metamodelos, proporcionando vinculaciones textuales o modelos de serialización para los mismos, y especificando la semántica operacional sobre la estructura de dichos modelos de serialización. Así mismo, se ha mostrado también cómo, tomando el núcleo lingüístico formado por la sintaxis abstracta y la semántica operacional como base, es posible articular los procesos de construcción de distintas herramientas, cada una dotada de un propósito bien definido (v.g. prototipado rápido, verificación de propiedades, edición gráfica, etc.).

5.1.3. Arquitectura Flexible para sistemas e-learning dirigidos por Lenguajes de Modelado Educativo

La tercera aportación de esta Tesis ha sido proponer una arquitectura general para sistemas e-learning, la propuesta <e-LD>, que adopta el concepto de EML como núcleo central. Esta arquitectura prevé:

- La incorporación explícita de los instructores al proceso de producción, mantenimiento y evolución de las aplicaciones e-learning, mediante la caracterización de los aspectos educativos en términos de notaciones usables, típicamente visuales, y cercanas a los dominios de experiencia de dichos instructores.

- La operacionalización de los diseños mediante su transformación a notaciones ejecutables. En particular, la propuesta <e-LD> contempla el uso de lenguajes de formalización de procesos de negocio para tal fin, aunque dicha propuesta es generalizable a cualquier otro tipo de notación (v.g. EMLs estandarizados, como IMS-LD).
- La reutilización de diseños pre-existentes aplicando, de nuevo, transformaciones.

De esta forma, un aspecto distintivo de esta propuesta es situar la reutilización a un nivel similar a los de la autoría y la operacionalización. Así mismo, la propuesta también identifica los aspectos de autoría y reutilización como los más costosos de llevar a cabo, así como los más críticos para facilitar la adopción efectiva de los EMLs por parte de la comunidad educativa.

5.1.4. Herramienta <e-LD> Author para la autoría y la reutilización de diseños educativos en IMS-LD

La cuarta aportación de esta Tesis queda reflejada en la cristalización de los objetivos planteados en la herramienta <e-LD> Author. Esta herramienta aborda los aspectos de autoría y reutilización de diseños educativos IMS-LD en <e-LD>. La racionalidad de la misma descansa en la complejidad que el nivel B de IMS-LD reviste de cara a la autoría directa de diseños, así como en la ausencia de herramientas disponibles que mitiguen dicha complejidad. Este hecho lleva a la clasificación básica de EMLs en *EMLs de autoría* y *EMLs de intercambio*, así como a la orquestación de la herramienta <e-LD> Author en base a dicha clasificación. Efectivamente, <e-LD> Author adopta una notación orientada a flujo como EML de autoría y propone la especificación IMS-LD como EML de intercambio. Así mismo, <e-LD> Author incluye soporte para la transformación entre ambas notaciones:

- Por una parte, el desarrollo de la herramienta evidencia cómo es posible transformar automáticamente diseños orientados a flujo en diseños IMS-LD equivalentes. De esta forma, la semántica de secuenciamiento en cada diseño orientado a flujo puede re-escribirse automáticamente en un diseño donde el secuenciamiento de actividades y otros elementos se lleva a cabo mediante reglas.
- Por otra parte, dicho desarrollo también ha evidenciado cómo la transformación inversa no puede automatizarse completamente. Sin embargo, sí es posible adoptar un enfoque semiautomático, mediante la inclusión de herramientas de soporte que asisten a los instructores (posiblemente apoyados por desarrolladores) en el proceso de reutilización. Más concretamente, en <e-LD> Author se recupera automáticamente el nivel A del diseño, mientras que el nivel B se utiliza para obtener representaciones explícitas de las dependencias entre los

distintos elementos: un grafo de dependencias que involucra propiedades, reglas, actividades, etc., y una vista hipertextual y por facetas del diseño.

De esta forma, <e-LD> Author:

- Facilita, por una parte, el uso de EMLs estandarizados (en particular, de IMS-LD) por parte de los instructores, ya que éstos pueden utilizar una notación de autoría mas sencilla y usable, en la que pueden expresar el secuenciamiento de actividades mediante un paradigma orientado a flujo de control (Grigoreanu et al., 2009). La especificación de la secuenciación de actividades mediante un diagrama de flujo de control, al estilo de los diagramas de actividades UML o de los diagramas de flujo típicos de los lenguajes de proceso de negocio, es mucho más sencilla que la descripción del secuenciamiento mediante un sistema global de reglas de producción, como el incluido en IMS-LD (Dodero et al., 2007). Así mismo, la notación visual de autoría adoptada en <e-LD> Author ofrece ventajas de usabilidad evidentes frente a la tersa sintaxis XML de IMS-LD o el reflejo directo de dicha sintaxis en interfaces basadas en formularios (v.g. RELOAD) que, sin embargo, no ocultan la complejidad inherente al nivel B del lenguaje.
- Facilita, por otra parte, el proceso de reutilización de diseños ya existentes. Efectivamente, la comprensión directa de un diseño IMS-LD es una tarea compleja, que se vé dificultada por los múltiples niveles de indirección propios del lenguaje, así como por las complejidades del secuenciamiento basado en reglas del mismo. Para ello, <e-LD> Author automatiza lo máximo posible el proceso de importación a un diseño orientado a flujo más comprensible. Además, <e-LD> proporciona herramientas complementarias para facilitar la comprensión del resto de los aspectos del diseño inicial.

Por tanto, la construcción de esta herramienta ejemplifica y demuestra la factibilidad de satisfacer los dos objetivos básicos planteados al inicio de esta Tesis.

5.2. Trabajo Futuro

Esta sección concluye el capítulo presentando algunas líneas futuras de investigación que se desprenden de esta Tesis. Más concretamente, se consideran las siguientes ocho líneas de investigación como las más prometedoras para continuar el trabajo iniciado en esta Tesis:

- Completar <e-LD> Author y mejorar su usabilidad.
- Inclusión del nivel C de IMS-LD en <e-LD> Author.
- Aplicación de la aproximación <e-LD> a otros lenguajes de intercambio.
- Integración de <e-LD> Author con motores de ejecución de IMS-LD.

- Mejora del enfoque de importación y reingeniería en <e-LD>.
- Creación de un *DSL Toolkit* para EMLs.
- Creación de un motor de ejecución IMS-LD basado en tecnologías de gestión de procesos de negocio.
- Aplicación al dominio de los juegos y las simulaciones educativas.

Los siguientes puntos motivan brevemente cada una de estas líneas de investigación y de trabajo futuro.

5.2.1. Completar <e-LD> Author y mejorar su usabilidad

El desarrollo de la herramienta de autoría <e-LD> Author desarrollada como parte de esta Tesis ha estado orientado a ejemplificar y contrastar los objetivos planteados en la misma: facilitar el uso de EMLs así como la reutilización de diseños educativos. De esta forma, el producto desarrollado debe considerarse más como un prototipo que como una herramienta final. Por tanto, una vez contrastada su utilidad, se plantea llevar a cabo un proceso de paso de prototipo a producto, mediante la refactorización de la herramienta. Asimismo, se debería realizar un proceso de mejora de la usabilidad de la misma en función de las pruebas realizadas con usuarios finales. El objetivo sería disponer de una herramienta madura y estable que se pueda distribuir de manera pública y como código abierto, a fin de que ésta pueda ser utilizada por una comunidad más amplia de educadores.

5.2.2. Inclusión del nivel C de IMS-LD en <e-LD> Author

En el trabajo llevado a cabo en esta Tesis se ha identificado el nivel B de IMS-LD como el de mayor complejidad de cara a la autoría de UoLs. De esta forma, en el desarrollo de <e-LD> se han obviado los detalles relativos al nivel C (notificaciones), dado que dicho nivel no añade características de secuenciamiento adicionales. No obstante, y por completitud, se estima oportuno abordar la inclusión de este nivel en futuras versiones de la herramienta.

5.2.3. Aplicación de la aproximación <e-LD> a otros lenguajes de intercambio.

En este trabajo se ha contrastado la propuesta <e-LD> tomando IMS-LD como principal caso de estudio. No obstante, el enfoque es perfectamente generalizable a otros EMLs de intercambio. En particular, se considera interesante aplicar la metodología propuesta en esta Tesis al lenguaje Simple Sequencing (SS), y, más concretamente, a su perfil de aplicación SCORM SN. Para ello, siguiendo el modelo de trabajo en etapas llevado a cabo en esta Tesis, puede comenzarse

realizándose un análisis y desarrollo inicial para generar plantillas de cursos SCORM SN, definiendo la estructura básica del curso y todas las características relativas a la navegación, pero no a la inclusión de los contenidos educativos. Estas plantillas podrán ser utilizadas como punto de partida para el desarrollo de diseños educativos basados en esta especificación, aliviando las dificultades técnicas y la necesidad de disponer de unos profundos conocimientos de dicha especificación (González-Barbone & Anido-Rifón, 2008). Así mismo, también se propone extender SS con características existentes en IMS-LD, siguiendo un enfoque similar al adoptado en las extensiones propuestas por la iniciativa LETSI (www.letsi.org), a fin de permitir crear diseños educativos colaborativos basados en dicha especificación.

5.2.4. Integración de <e-LD> Author con motores de ejecución de IMS-LD.

La versión actual de <e-LD> consiste en una herramienta independiente, a partir de la cuál es posible generar paquetes de intercambio IMS cuya sección de organización se ajusta a IMS-LD. Estos paquetes pueden, posteriormente, incorporarse de forma manual a sistemas compatibles con IMS-LD. No obstante, se considera interesante facilitar la tarea de despliegue e instanciación de las UoLs mediante la interconexión de <e-LD> con distintos motores de ejecución de IMS-LD. Así mismo, dicha integración tendrá también un impacto positivo a nivel de autoría, al permitir aprovechar, por ejemplo, elementos dependientes de la plataforma de despliegue (v.g. propiedades y roles globales). En este aspecto se podría tratar de integrar con el motor de ejecución de IMS-LD desarrollado en la Universidad Carlos III de Madrid por el grupo GAST (Escobedo et al., 2007).

5.2.5. Mejora del enfoque de importación y reingeniería en <e-LD>.

En esta Tesis se ha puesto de manifiesto las interrelaciones existentes entre el problema de reutilización de diseños educativos con escenarios de reingeniería (y, más concretamente, de ingeniería inversa) en los campos de desarrollo de Software y de Ingeniería de Procesos de Negocio. De esta forma, y como extensión lógica del enfoque dirigido por modelos adoptado en la definición de la notación de autoría de <e-LD>, se considera interesante investigar la aplicabilidad de las modernas tecnologías de reingeniería y modernización de software al problema de importación y reingeniería en <e-LD>. En particular, se propone analizar la aplicabilidad de la propuesta *Architecture-Driven Modernization* (ADM) de OMG para tal fin (Ulrich, 2007).

5.2.6. Creación de un *DSL Toolkit* para EMLs.

Utilizando como base la experiencia adquirida en el desarrollo del prototipo de <e-LD> Author, y aprovechando la evolución de las tecnologías subyacentes, tales como el Eclipse Modeling Project (Gronback, 2009) o las plataformas de desarrollo de Software dirigidas por

lenguajes (Fowler, 2010), como trabajo futuro de esta Tesis se quiere explorar la creación de un conjunto de herramientas y librerías que pueda ser utilizado como base para la generación de otras herramientas de soporte específicas para la autoría y reingeniería de diseños educativos formalizados con EMLs. Estos componentes proporcionarán un andamiaje metalingüístico apropiado para llevar a cabo la construcción dirigida por EMLs de aplicaciones e-learning, y servirán como base para simplificar el uso práctico del modelo de proceso propuesto en esta Tesis.

5.2.7. Creación de un motor de ejecución IMS-LD basado en tecnologías de gestión de procesos de negocio.

En esta Tesis se ha indicado el paralelismo existente entre los diseños educativos orientados a actividades y los modelos de proceso de negocio. De esta forma, se considera interesante estudiar cómo transformar IMS-LD a distintos lenguajes de modelado de procesos de negocio, a fin de permitir ejecutar UoLs reutilizando motores de ejecución de flujos de trabajo ya existentes. En la realización de esta experiencia podrán reutilizarse iniciativas como *Process Virtual Machine* (Baeyens & Valdes Faura, 2007), que abordan la problemática de ejecutar diferentes notaciones de modelado de procesos de negocio utilizando una infraestructura común.

5.2.8. Aplicación al Dominio de los Juegos y las Simulaciones Educativas

Adicionalmente a las líneas de investigación futuras descritas anteriormente, algunos aspectos del trabajo de esta Tesis han servido como ideas seminales para el planteamiento de dos Tesis Doctorales actualmente en curso en el campo de los juegos y las simulaciones educativas:

- Siguiendo las ideas de mantener la compatibilidad con los estándares existentes en la comunidad educativa, el doctorando D. Ángel del Blanco Aguado persigue el objetivo de integrar sistemas interactivos complejos, como los juegos educativos, dentro de las plataformas de gestión del aprendizaje utilizando una aproximación dirigida por lenguajes estandarizados.
- En torno a las ideas de crear lenguajes de autoría específicos que simplifican la participación y la integración de los expertos en el dominio durante el proceso de creación de los lenguajes específicos y las herramientas de soporte, el doctorando D. Eugenio Jorge Marchiori persigue aplicar estas ideas en el campo de la autoría y creación de juegos y simulaciones educativas.

Capítulo 6

Artículos Presentados

A continuación se incluyen los artículos editados que se aportan como parte de esta Tesis Doctoral.

6.1. Educational Modeling Languages: A Conceptual Introduction and a High-Level Classification

Cita completa:

Martinez-Ortiz I, Moreno-Ger P, Sierra JL and Fernandez-Manjon B. Educational Modeling Languages. A Conceptual Introduction and a High-Level Classification. En *Computers and Education E-Learning, From Theory to Practice*. Dordrecht, Países Bajos: Springer. p. 27-40. ISBN 978-1-4020-4913-2 (libro impreso) ISBN 978-1-4020-4914-9 (libro digital).

Resumen original de la contribución

Creating good quality learning resources is not sufficient for an optimum learning experience. Equally important is having a more enabling learning process involving not only the delivery of learning materials but also other activities that the learner must carry out to meet the learning objectives proposed by the instructor (tutoring, tests, reading books, etc.). Educational Modeling Languages (EMLs) are the cornerstone of e-learning because they provide a language that can be used by the instructors to formalize their own teaching process so that it can also be interpreted by computers. In this chapter we provide a conceptual introduction and a high-level classification of some of the proposed EMLs.

Referencia de citas bibliográficas

(ADL, 2004); (AICC, 2006); (AICC/CMI_CMI001, 2004); (ARIADNE, 2006); (Avgeriou et al., 2003); (Barrett-Baxendale et al. 2005); (Botturi, 2006); (Brickley, 1995); (Buendía-García & Díaz-Perez, 2003); (Dalziel, 2003); (Dick et al., 2000); (Downes, 2001); (Dumas et al., 2005); (Durm et al., 2001); (Duval et al., 2001); (Fernandez-Manjon and Sancho, 2002); (Gamma et al., 1994); (IEEE, 2002); (IEEE, 2006); (IMS, 2002); (IMS, 2003); (IMS, 2003b); (IMS, 2006); (IMS, 2006b); (Koch, 2002); (Koper, 2000); (Koper, 2001); (Koper, 2003); (LAMS, 2005); (Martens & Vogten, 2005); (Martinez-Ortiz et al., 2005); (Martinez-Ortiz et al., 2006); (Merrill, 1994); (Moreno-Ger et al., 2005); (OUNL, 2005); (Paquette et al., 1997); (Paquette, 2001); (Paquette, 2004); (Paquette et al., 2005); (Rawlings et al., 2002); (Reigeluth, 1983); (Rodriguez-Artacho et al., 1999); (Slavin, 1995); (Vantroys, 2003); (Verbert & Duval, 2004); (Walsh & Muellner, 1999); (Weitl et al., 2002).

6.2. Supporting the Authoring and Operationalization of Educational Modelling Languages

Cita completa:

Martínez-Ortiz I, Moreno-Ger P, Sierra-Rodríguez JL, Fernández-Manjón B. Supporting Authoring and Operationalization of Educational Modelling Languages. *Journal of Universal Computer Science*; 28 de Julio de 2007; 13(7): 938-947.

Resumen original de la contribución

The modelling of educational processes and their operational support is a key aspect in the construction of more effective e-learning applications. Instructional models are usually described by means of an *educational modelling language* (EML). The EML used can be one of the available standards (e.g. IMS Learning Design), the customization of a standard to meet a specific application profile, or even a domain-specific EML specifically designed to better fit the very particular needs of a learning scenario. In this paper we present <e-LD>, a general authoring and operationalization architecture capable of dealing with all these possibilities in a highly modular and flexible way. We also outline a specific implementation of <e-LD> based on standard XML technologies and workflow management systems, and we describe how this implementation can be used to support IMS Learning Design.

Referencia de citas bibliográficas

(Anane et al. 2005); (Andrews et al. 2003); (Dumas et al., 2005); (Gibson, 2003); (IMS, 2003); (Kim et al. 2005); (Koper, 2001); (Koper & Olivier, 2004); (Martínez-Ortiz et al. 2006b); (Martínez-Ortiz et al. 2007); (OMG 2005); (OMG, 2005b); (Paquette et al., 2005b); (Polsani, 2003); (Sierra et al., 2004); (Sierra et al., 2005); (Sierra et al., 2005b); (Sierra et al., 2005c); (Sierra, et al., 2006); (Sierra et al., 2007b); (Vantroys & Peter, 2003).

6.3. Enhancing Reusability of IMS-LD Units of Learning: The e-LD Approach

Cita completa:

Martínez-Ortiz I, Sierra JL, Fernández-Manjón B. Enhancing Reusability of IMS-LD Units of Learning: The e-LD Approach. *8th IEEE International Conference on Advanced Learning Technologies (ICALT 2008)*. 1-5 de Julio de 2008; Santander, España. Washington DC, USA: IEEE Computer Society; 2008. pp 402-404.

Resumen original de la contribución

In this paper we describe the e-LD approach for the design and repurposing of Units of Learning (UoLs). This approach is centered in domain-specific Educational Modeling Languages (EMLs) built with the close collaboration between instructors and developers. The products of this collaboration are: (i) the definition of a suitable authoring EML, and (ii) the construction of software tools to allow the importation and authoring of UoLs. The domain-specific authoring EMLs and tools simplify the production and repurposing of UoLs.

Referencias de citas bibliográficas

(ADL, 2006); (Brownston et al., 1985); (Clayberg & Rubel, 2006); (IMS, 2003); (IMS, 2003b); (Martínez-Ortiz et al., 2007b); (Stahl et al., 2006).

6.4. Language Engineering Techniques for the Development of E-Learning Applications

Cita completa:

Martinez-Ortiz, I., Sierra, J.L., Fernandez-Manjon, B., Fernandez-Valmayor, A., Language engineering techniques for the development of e-learning applications, *Journal of Network and Computer Applications*, Septiembre de 2009, 32(5): 1092-1105. ISSN 1084-8045DOI: 10.1016/j.jnca.2009.02.005.

Resumen original de la contribución

In this paper we propose the use of language engineering techniques to improve and systematize the development of e-learning applications. E-learning specifications usually rely on domain-specific languages that describe different aspects of such final e-learning applications. This fact makes it natural to adopt well-established language engineering principles during the construction of these applications. These principles promote the specification of the structure and the runtime behavior of the domain-specific languages as the central part of the development process. This specification can be used to drive different activities: rapid prototyping, provision of authoring notations and tools, automatic model checking of properties, importation/exportation from/to standards, and deployment of running applications. This language engineering concept also promotes active collaboration between instructors (the users of the languages) and developers (the designers and implementers) throughout the development process. In this paper we describe this language-driven approach to the construction of e-learning applications and we illustrate all its aspects using a learning flow sequencing language as a case-study.

Referencias de citas bibliográficas

(Aalst & Kees, 2004); (Aho et al., 2006); (Allen, 2006); (Baldoni et al., 2007); (Baldoni & Marengo, 2007); (Birbeck et al., 2001); (Booch et al., 1998); (Bork, 1985); (Burch et al., 1992); (Burgos et al., 2007); (Burgos et al., 2007b); (Cimatti et al., 2000); (Clarke et al., 1986); (Clarke et al., 2000); (Clément et al., 1986); (Comon et al., 2007); (Deursen et al., 2000); (Emerson, 1990); (Fernández-Manjón and Fernández-Valmayor, 1997); (Friedman et al., 2001); (Friesen, 2005); (Fuentes-Fernández et al., 2007); (IMS, 2004); (IMS, 2005); (Goodwill & Hightower, 2004); (IMS, 2003); (IMS, 2003b); (IMS, 2004); (IMS, 2005); (IMS, 2006c); (Klint et al., 2005); (Koper & Manderveld, 2004); (Koper & Olivier, 2004); (Krasner & Pope, 1988); (Makoto et al., 2005); (Marriott et al., 1999); (Martínez-Ortiz et al. 2007); (Martínez-Ortiz et al., 2007b); (Martínez-Ortiz

et al., 2008); (Martínez-Ortiz et al., 2008b); (Martínez-Ortiz et al., 2009); (Mernik et al., 2005); (Molenda, 2003); (Moreno-Ger et al., 2006); (Moreno-Ger et al., 2007); (Moreno-Ger et al., 2009); (Mosses, 2004); (Mosses, 2006); (Paakki, 1995); (Parr, 2007); (Pereira & Warren, 1980); (Plotkin, 2004); (Sarasa-Cabezuelo et al., 2008); (Sierra, et al., 2006); (Sierra et al., 2006b); (Sierra et al., 2006c) ; (Sierra & Fernández-Valmayor, 2007); (Sierra et al., 2007) ; (Sierra et al., 2007b); (Sierra et al., 2007c); (Sierra et al., 2008); (Sierra et al., 2008b); (Stahl et al., 2006); (Stanchfield, 2009); (Steinber et al., 2003); (Sterling & Saphiro, 1994).

6.5. Authoring and Reengineering of IMS Learning Design Units of Learning

Cita completa:

Martínez-Ortiz I., Sierra J.L., Fernández-Manjón B. Authoring and Reengineering of IMS Learning Design Units of Learning. *IEEE Transactions on Learning Technologies* 2(3) , Marzo 2009: 189-202. IEEE computer Society Digital Library. IEEE Computer Society, DOI: 10.1109/TLT.2009.14.

Resumen original de la contribución

Educational Modeling Languages (EMLs) are notations that allow instructors to formally describe educational processes, including teaching and learning interactions and activities. The description of a specific teaching process using an EML is called a learning design. EMLs, where IMS Learning Design (IMS-LD) is becoming a “de facto” standard, address aspects such as the interoperability and reusability of teaching practices across learning management systems. However, the actual application of EMLs is being hindered by different problems such as the technical skills required to use typical EMLs and the difficulty of understanding and maintaining preexisting learning designs. Thus, to promote the adoption of EMLs, it is necessary to provide more user-friendly tools and methodologies to facilitate their assimilation and reduce the workload required to use them. In this paper, we present the e-LD system, which provides: a graphical notation to design or redesign learning designs, an import-modification-export process to reengineer IMS-LD learning designs and a tool to generate and analyze dependencies between different IMS-LD elements.

Referencias de citas bibliográficas

(Aalst & Kees, 2004); (ADL, 2006); (Aguilar-Saven, 2004); (Botturi, 2006); (Botturi et. al., 2006); (Brownston et al., 1985); (Burgos, 2005); (Caeiro et. al., 2007); (Chen, 1985); (Dalziel, 2003); (Dick et al., 2000); (Ferrucci et. al., 2002); (Griffiths et. al., 2005); (Griffiths et al., 2008); (Hernández et. al., 2007); (Heyer et. al., 2007); (IMS, 2003); (IMS, 2004); (Karampiperis & Sampson, 2004); (Koper & Burgos, 2005); (Koper & Tattersall, 2005); (Laforcade, 2007); (Li, 1991); (Martínez-Ortiz et al., 2007b); (Martínez-Ortiz et al., 2008b); (Martínez-Ortiz et al., 2009); (Martínez-Ortiz et. al., 2009b); (Miao, 2005); (Milligan et al., 2005); (Muñoz Merino et. al., 2009); (Ouyang et. al., 2006); (Paquette et al., 2006); (Polsani, 2003); (Recker & Mendling, 2006); (Sloep, 2004); (Sloep et. al., 2005); (Weller et. al., 2006).

6.6. Using DocBook and XML Technologies to Create Adaptive Learning Content in Technical Domains

Cita completa:

Martínez-Ortiz I., Moreno-Ger P., Sierra J.L., Fernández-Manjón B. Using DocBook and XML Technologies to Create Adaptive Learning Content. *International Journal of Computer Science and Applications*; June 2006; 3(2): 91-108.

Resumen original de la contribución

This work presents an XML-based authoring methodology that facilitates the different tasks associated with the development of standards-compliant e-learning content development. The idea is to support a unified authoring process of adaptive e-learning content by adopting the manual writing metaphor. According to this metaphor, which is specially well-suited for technical domains, instructors are compelled to view e-learning material as the manuals normally written and used to teach a particular subject. The methodology is supported by the DocBook document model (an XML application) which has a number of tools associated that facilitate the implementation of the methodology. Other tasks such as the packaging of contents or publishing in the Learning Management System (LMS) can be automated from these manuals.

Referencias de citas bibliográficas

(Amorim et al., 2003); (Baudry et al., 2004); (Birbeck et al., 2001); (Bungenstock et al., 2002); (Boticario et al., 2003); (Coombs et al., 1987); (Dahn, 2002); (Dolan et al., 2005); (Dougherty, 1999); (Downes, 2001); (Farrel et al., 2004); (Goldfarb, 1981); (Gueye et al., 2004); (González-Barahona et al., 2005); (González-Barahona et al., 2005b); (IEEE, 2002); (IMS, 2004); (Koper, 2003); (Koper & Manderveld, 2004); (Koper & Olivier, 2004); (McMartin, 2004); (Molloy, 2003); (Polsani, 2003); (Rathz et al., 2004); (Sierra et al., 2004); (Sierra et al., 2005b); (Stayton, 2005); (Walsh, 1998); (Walsh, 2001); (Walsh, 2004); (Walsh & Muellner, 1999); (W3C, 1999); (W3C, 2001).

6.7. Enhancing IMS-LD Units of Learning Comprehension

Cita completa:

Martínez-Ortiz I., Sierra J.L., Fernández-Manjón B. *Enhancing IMS-LD Units of Learning Comprehension*. 4th International Conference on Internet and Web Applications and Services (ICIW 2009). 24-28 de Mayo de 2009; Venecia, Italia. pp 561-566. DOI 10.1109/ICIW.2009.91.

Resumen original de la contribución

IMS Learning Design (IMS-LD) is a powerful and expressive educational modeling language, which is becoming a “de facto” encoding and interchange standard for activity-based courses. IMS-LD expressivity includes functionalities such as the adaptation of the runtime behavior of the courses according to the performance and the specific needs of the learners. But these formally described courses, called Units of Learning (UoLs) in IMS-LD, are difficult to understand and reuse by instructors, hindering IMS-LD adoption. In this paper we describe how these shortcomings are resolved in e-LD, a flow oriented authoring tool for IMS-LD. In e-LD preexisting designs encoded in IMS-LD are automatically processed to produce a more understandable presentation with different views, such as a browse-able hyper-textual view or a graphical representation of the dependencies between UoL conditions and other components of the learning design. These views help to enhance UoLs’ comprehension by instructors.

Referencias de citas bibliográficas

(Aalst & Kees, 2004); (ADL, 2006); (Booch et al., 2005); (Burgos et al., 2007); (Dalziel, 2006); (Griffiths et. al., 2005); (Griffiths et al., 2008); (Heyer et. al., 2007); (IMS, 2003); (Koper & Tattersall, 2005); (Knuth, 1984); (Li, 1991); (Martínez-Ortiz et al. 2007); (Martínez-Ortiz et al., 2007b); (Martínez-Ortiz et al., 2008); (Martínez-Ortiz et al., 2009); (Mayrhauser & Vans, 1995); (Milligan et al., 2005); (Paquette et al., 2006); (Ramaswamy et al., 1997); (Specht & Burgos, 2007);

6.8. Translating e-learning Flow-Oriented Activity Sequencing Descriptions into Rule-based Designs

Cita completa:

Martínez-Ortiz I, Sierra JL, Fernández-Manjón B. Translating e-learning Flow-Oriented Activity Sequencing Descriptions into Rule-based Designs. Proceedings of the 6th International Conference on Information Technology: New Generations (ITNG 2009). 27-29 de Abril de 2009; Las Vegas, EEUU. Washington DC, USA: IEEE Computer Society; 2009. pp. 1108-1113. DOI: 10.1109/ITNG.2009.125.

Resumen original de la contribución

In this paper, we describe how to automatically translate e-learning flow-oriented activity sequences into rule-based designs, such as those supported by the “de-facto” e-learning modeling standard: the IMS Learning Design specification. Our aim is that instructors model their educational designs using a user-friendly visual notation. Then these designs can be automatically exported into standardized and interoperable representations, which can be interchanged with / deployed in a plethora of heterogeneous Learning Management Systems and tools. This approach has been implemented in e-LD, an authoring system which supports the authoring and refactoring of IMS Learning Designs using a flow-oriented visual syntax.

Referencias de citas bibliográficas

(Aho et al., 2006); (Burgos et al., 2007); (Caeiro et. al., 2007); (Cutland, 1980); (Dalziel, 2006); (Dodero et al., 2007); (Ibrahim, 1989); (IMS, 2003); (IMS, 2003b); (Koper & Tattersall, 2005); (Laforcade, 2005); (Li, 1991); (Marriott et al., 1999); (Martínez-Ortiz et al. 2007); (Martínez-Ortiz et al., 2007b); (Martínez-Ortiz et al., 2008); (Specht & Burgos, 2007).

Bibliografía

- (Aalst & Kees, 2004) Aalst W. & Kees H. (2004). *Workflow Management: Models, Methods, and Systems*. Massachusetts: MIT Press.
- (ADL, 2004) Advanced Distributed Learning (2004). *SCORM 2004 Overview 2nd Edition Version 1.0*.
- (ADL, 2006) Advanced Distributed Learning (2006). *Shareable Content Object Reference Model (SCORM) 2004 3rd Edition Sequencing and Navigation Version 1.0*.
- (ADL, 2009) Advanced Distributed Learning (2009). *SCORM 2004 Overview 4th Edition Version 1.0*.
- (Agostinho et. al., 2008) Agostinho, S., Harper, B. M., Oliver, R., Hedberg, J., & Wills, S. (2008). A visual learning design representation to facilitate dissemination and reuse of innovative pedagogical strategies in university teaching. In L. Botturi & S. Todd. Stubbs (Eds.), *Handbook of Visual Languages for Instructional Design: Theories and Practices* (pp. 380-393). Hershey PA: Information Science Reference, IGI Global.
- (Agostinho, 2009) Agostinho, S. (2009). Learning Design representations to document, model and share teaching practice. In L. Lockyer, S. Bennett, S. Agostinho, & B. Harper (Eds.). *Handbook of Research on Learning Design and Learning Objects: Issues, Applications and Technologies* (pp. 1-19). Information Science Reference.
- (Aguilar-Saven, 2004) Aguilar-Saven, R.S. (2004). Business process modelling: Review and framework. *Int. Journal of Production Economics*, 90(2), 129-149.
- (Aho et al., 2006) Aho A.V., Lam M.S., Sethi R. & Ullman JD. (2006). *Compilers: principles, techniques and tools* (2nd ed). Boston, MA, USA: Addison-Wesley.
- (AICC/CMI_CMI001, 2004) Aviation Industry CBT Committee - AICC Subcommittee (2004). CMI Guidelines for Interoperability AICC revision 4.0.
- (AICC, 2006) Aviation Industry CBT Committee (AICC). Disponible online en: <http://www.aicc.org> (último acceso 2 de Noviembre de 2010).
- (Allen, 2006) Allen, C.W. (2006). Overview and Evolution of the ADDIE Training System. *Advances in Developing Human Resources*, 8(4): 430-441.

- (Alves et al., 2007) Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Goland, Y., Guízar, A., Kartha, N., Liu, C. K., Khalaf, R., König, D., Marin, M., Mehta, V., Thatte, S., van der Rijn, D., Yendluri, P., Yiu, A. (2007). *Web Services Business Process Execution Language Version 2.0*. OASIS Standard. Disponible online: <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html> (último acceso 2 de Noviembre de 2010).
- (Ambler, 2003) Ambler, S. W., (2003). Agile Modeling (AM) Home Page: Effective Practices for Modeling and Documentation. Disponible online en: <http://www.agilemodeling.com/essays/amdd.htm> (último acceso 2 de Noviembre de 2010).
- (Ambler, 2004) Ambler, S. W. (2004). *The Object Primer: Agile Model-Driven Development with UML 2.0*. Cambridge University Press.
- (Amorim et al., 2003) Amorim, R., Lama, M., Sánchez, E. & Vila, X.A. (2003). An Educational Ontology based on Metadata Standards. *European Conference on eLearning*. Glasgow, UK. pp. 29-36.
- (Anane et al. 2005) Anane, R., Bordbar, B., Deng F. & Hendley, R. J (2005). A Web services approach to learning path composition. Actas del congreso *IEEE International Conference on Advanced Technologies (ICALT 2005)* (pp. 98-102).
- (Andrews et al. 2003) Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I. & Weerawarana, S. (2003). *Business Process Execution Language for Web Services Version 1.1*. IBM, BEA Systems, Microsoft, SAP AG, Siebel Systems Specification.
- (ARIADNE, 2006) Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE) (2006). Disponible online en: <http://www.ariadne-eu.org>.
- (Avgeriou et al., 2003) Avgeriou, P., A. Papasalouros, Retalys, S. & Skordalakis, M. (2003). Towards a Pattern Language for Learning Management Systems. *Educational Technology & Society*, 6(2), 11-24.
- (Baeyens & Valdes Faura, 2007) Baeyens, T. & Valdes Faura, M. (Mayo, 2007) The Process Virtual Machine. Disponible online en: <http://docs.jboss.com/jbpm/pvm/article/> (último acceso 2 de Noviembre de 2010).
- (Baldoni et al., 2007) Baldoni M., Baroglio C., Brunkhorst I., Marengo E. & Patti, V. (2007). Reasoning-Based Curriculum Sequencing and Validation: Integration in a Service-Oriented Architecture. In E. Duval, R. Klamma, M. Wolpers (Eds.), *Creating New Learning Experiences on a Global Scale*. 2nd European Conference on Technology Enhanced Learning (EC TEL 2007) (pp. 426-431). Berlin: Springer.
- (Baldoni & Marengo, 2007) Baldoni M. & Marengo E. (2007). Curriculum Model Checking: Declarative Representation and Verification of Properties. In E. Duval, R. Klamma, M.

- Wolpers (Eds.), *Creating New Learning Experiences on a Global Scale*. 2nd European Conference on Technology Enhanced Learning (EC TEL 2007) (pp.432-437). Berlin: Springer.
- (Barr, A., 2010) Barr, A. *Beyond Content Portability: Shifting Gears in Elearning Innovation*. LETSI Technical Roadmap. Disponible online: <http://wiki.letsi.org/download/attachments/6324974/Beyond+Content+Portability+9June10.pdf?version=1&modificationDate=1278977658000> (última visita: 12 de Diciembre de 2010).
- (Barr, A., 2010b) Barr, A. *LETSI A Service Oriented Elearning Architecture: LETSI's Vision of the Future of CMI*. LETSI Technical Roadmap. Disponible online: <http://wiki.letsi.org/download/attachments/6324974/LETSI+CMI+Position+statement+27Aug10.pdf?version=1&modificationDate=1283129488000> (última visita: 12 de Diciembre de 2010).
- (Barrett-Baxendale et al. 2005) Barrett-Baxendale, M., Hazlewood, P. & Anderson, M. (2005). *SliDe, SLeD Demonstrator - Final Report v 1.0*. Hope Park, Liverpool Hope University.
- (Barreto et al., 2007) Barreto, C., Bullard, V., Erl, T., Evdemon, J., Jordan, D., Kand, K., König, D., Moser, S. Stout, R., Ten-Hove, R., Trickovic, I., van der Rijn, D. & Yiu, A. (2007) *Web Services Business Process Execution Language Version 2.0: Primer*. Disponible online en: <http://www.oasis-open.org/committees/download.php/23964/wsbpel-v2.0-primer.htm> (último acceso 27 de Octubre de 2.010).
- (Basu et al., 1997) Basu, A., Hayden, M., Morrisett, G. & von Eicken, T. (Enero, 1997). A Language-Based Approach to Protocol Construction. First *ACM SIGPLAN Workshop on Domain-Specific Languages DSL '97*.
- (Baudry et al., 2004) Baudry, A., Bungenstock, M. & Mertsching, B. (2004). Reusing Document Formats for Modular Course Development. World Conference on Educational Multimedia, Hypermedia & Telecommunications, ED-MEDIA (pp. 535-537).
- (Beaulieu, 2005) Beaulieu, A. (2005). *Learning SQL*. O'Reilly Media.
- (Beetham, 2004) Beetham, H. (2004) *Review: developing e-Learning Models for the JISC Practitioner Communities*. Version 2.1. Joint Information Systems Committee e-learning and Pedagogy Programme.
- (Bentley, 1986) Bentley, J.L. (1986). Programming Pearls: Little languages. *Communications of the ACM*, 29(8), 711-721.
- (Berggreen et al., 2005) Berggreen, A., Burgos, D., Fontana, J. M., Hinkelman, D., Hung, V., Hursh, A. & G. Tielemans, G. (2005). Practical and pedagogical issues for teacher adoption of ims learning design standards in moodle LMS. *Journal of Interactive Media in Education*, 2005(2), 1-24.
- (Berlanga & García, 2005) Berlanga, A.J. & García, F.J. (2005). IMS-LD reusable elements for adaptive learning designs. *Journal of Interactive Media in Education* 2005(2), 1- 16.

- (Birbeck et al., 2001) Birbeck M., Kay M., Livingstone S., Mohr S.F., Pinnock J. & Loesgen B. (2001) *Professional XML* (2nd edition). Birmingham: Wrox Press.
- (Booch et al., 1998) Booch, G., Rumbaugh, J. & Jacobson I. (1998). *The Unified Modeling Language User Guide*. Reading, Massachusetts: Addison Wesley.
- (Booch et al., 2005) Booch, G., Rumbaugh, J. & Jacobson, I. (2005). *The Unified Modeling Language User Guide* (2nd Edition). Reading, Massachusetts: Addison-Wesley.
- (Bork, 1985) Bork, A. (1985). *Personal Computers for Education*. New York, NY, USA: Harper & Row Publishers, Inc.
- (Boticario et al., 2003) Boticario J. G., Raffenne, E., Aguado, M., Arroyo, D., Cordova, M. A., Guzmán, J.L., Garcia, T., Hermira, S. Ortiz, J., Pesquera, A., Romojaro, H. & Valiente., S. (2003). The INNOVA Approach. *EADTU Annual Conference: E-Bologna*.
- (Botturi, 2006) Botturi, L. (2006). E2ML: A Visual Language for the Design of Instruction. *Educational Technology Research and Development*, 54(3), 265-293.
- (Botturi et. al., 2006) Botturi, L., Derntl, M., Boot, E. & Figl, K. (2006). A Classification Framework for Educational Modeling Languages in Instructional Design. *Actas del congreso 6th IEEE International Conference on Advanced Learning Technologies (ICALT 06)* (pp. 1216-1220).
- (Botturi et al., 2008) Botturi, L., Burgos, D., Caeiro, M., Derntl, M., Koper, R., Parrish, P., Sodhi, T. & Tattersall, C. (2008). Comparing visual instructional design languages: a case study. In L. Botturi & T. Stubbs (Eds.), *Handbook of visual languages for instructional design: theories and practices* (pp. 315-343).
- (Brickley, 1995) Brickley, D. (1995). *Towards an open question-interchange framework*. Disponible online en: <http://www.ilrt.bris.ac.uk/netquest/about/lang/motivation.html> (último acceso Junio de 2006).
- (Britain, 2004) Britain, S. (2004). *A review of learning design: Concept, specification and tools*. JICS E-learning Pedagogy Programme report. Disponible online en: www.jisc.ac.uk/uploaded_documents/ACF83C.doc (último acceso Junio de 2010).
- (Bruce, 1997) Bruce, D. (1997). What Makes a Good Domain-Specific Language?: APOSTLE, and its Approach to Parallel Discrete Event Simulation. *First ACM SIGPLAN Workshop on Domain-Specific Languages DSL '97*.
- (Brownston et al., 1985) Brownston, L., Farrell, R., Kant, E. & Martin, N. (1985). *Programming Experts Systems in OPS5: An Introduction to Rule-based Programming*. Addison-Wesley.
- (Buendía-García & Díaz-Perez, 2003) Buendía-García, F. & P. Díaz-Perez, P. (2003). A Framework for the Management of Digital Educational Contents Conjugating Instructional and Technical Issues. *Educational Technology & Society*, 6(4), 48-59.

- (Buendía-García et al., 2004) Buendía-García, F., Agustí, F., Benlloch, J. V., Bisbal, E. & Llesma, M. (2004). XEDU, a proposal of learning management system implementation. *Journal of Information Technology Impact*, 4(1), 1-12.
- (Bungenstock et al., 2002) Bungenstock, M., Baudry, A. & Mertsching, B. (2002). The Construction Kit Metaphor for a Software Engineering Design of an E-Learning System. *World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 216-217).
- (Burch et al., 1992) Burch J.R., Clarke E.M., McMillan K.L., Dill D.L., Hwang L.J. (1992) Symbolic Model Checking: 10^{20} States and Beyond. *Information and Computation*, 98(2), 142-170.
- (Burgos, 2005) Burgos, D., (2005) Geo-Quiz 3, Disponible online en: <http://hdl.handle.net/1820/404> (último acceso Junio 2009).
- (Burgos & Griffiths, 2005) Burgos, D. & Griffiths, D. (2005) *The Unfold project. understanding and using learning design*. Open University of The Netherlands, Herleen.
- (Burgos et al., 2007) Burgos, D., Moreno-Ger, P., Sierra, J.L., Fernández-Manjón, B. & Koper, R. (2007). Authoring Game-Based Adaptive Units of Learning with IMS Learning Design and <e Adventure>. *International Journal of Learning Technology*, 3(3), 252-268.
- (Burgos et al., 2007b) Burgos, D., Tattersall, C. & Koper, R. (2007). Representing adaptive and adaptable Units of Learning: How to model personalized eLearning in IMS Learning Design. In B. Fernández-Manjón, J.M. Sánchez-Pérez, J.A. Gómez-Pulido JA, M.A. Vega-Rodríguez, J. Bravo-Rodríguez (Eds.), *Computers and Education: E-learning - from theory to practice* (pp. 41-56). Berlin: Springer.
- (Caeiro et al., 2007) Caeiro, M. Marcelino, M.J., Llamas, M., Anido-Rifón, L. & Mendes, A.J. (2007), Supporting the Modeling of Flexible Educational Units PoEML: A Separation of Concerns Approach. *Journal of Universal Computer Science*, 13(7) 980-990.
- (Caeiro Rodriguez et al., 2010) Caeiro Rogriguez, M., Derntl, M. & Botturi, L. (Diciembre, 2010) . Visual instructional design languages. *Journal of Visual Languages & Computing*, 21(6), 311-312.
- (Cerami, 2002) Cerami, E., (2002) *Web Services Essentials*. O'Reilly.
- (Chen, 1985) Chen, P. (1985). Database Design Using Entities and Relationship. In S.B. Yao (Ed.), *Principles of Data Base Design* (pp. 174-210). Prentice-Hall.
- (Cimatti et al., 2000) Cimatti, A., Clarker, E., Giunchiglia, F. & Roveri, M. (Marzo, 2000). NUSMV: a new symbolic model checker. *International Journal on Software Tools for Technology Transfer*. 2(4), 410-425.

- (Clarke et al., 1986) Clarke, E.M., Emerson, E.A. & Sistla, A.P. (Abril, 1986). Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. *ACM Transactions on Programming Languages and Systems*. 8(2), 244-263.
- (Clarke et al., 2000) Clarke, E.M., Grumberg, O. & Peled, D.A. (2000). *Model Checking*. Massachusetts: MIT Press.
- (Clayberg & Rubel, 2006) Clayberg, E. & Rubel, D. (2006). *Eclipse: Building Commercial-Quality Plug-ins* (2nd Edition). Addison Wesley Professional.
- (Cleaveland, 1988) Cleaveland, J.C. (1988). Building Application Generators. *IEEE Software* 5(4), 25-33.
- (Cleaveland, 2001) Cleaveland, J.C. (2001). *Program Generators with XML and Java*. Prentice Hall.
- (Clément et al., 1986) Clément, D., Despeyroux, J., Despeyroux, T., Hascoet, L. & Kahn, G. (1986). Natural Semantics on the Computer. In: Fuchi K, Nivat M, editors, *Actas del congreso France-Japan AI and CS Symposium* (pp. 49-89).
- (Comon et al., 2007) Comon, H., Dauchet, M., Gilleron, R., Jacquemanrd, F., Lugiez, D., Löding, C., Tison, S. & Tommasi, M. (2007). *Tree Automata Techniques and Applications*. Disponible online en <http://tata.gforge.inria.fr/> (ultimo acceso Enero 2009).
- (Coombs et al., 1987) Coombs, J., Renear, A. & DeRose S. (1987). Markup Systems and the Future of Scholarly Text Processing, *Communications of the ACM*, 30(11), 933-947.
- (Cook et al., 2007) Cook, S., Jones, G., Kent, S. & Cameron, A. (2007). *Domain-Specific Development with Visual Studio DSL Tools*. Addison-Wesley.
- (Crew, 1997) Crew, R.F. (Octubre, 1997). ASTLOG: A Language for Examining Abstract Syntax Trees. Actas de la conferencia *USENIX Conference on Domain-Specific Languages*. Berkely, CA.
- (Curbera et al., 2002) Curbera, F., Goland, Y., Klein, J., Leymann, F., Roller, D., Thatte, S. & Weerawarana, S. (Julio, 2002). *Business process execution language for web services, version 1.0*. specification, BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG, Siebel Systems.
- (Cutland, 1980) Cutland, N.J. (1980). *Computability: An Introduction to Recursive Function Theory*. Cambridge Univ. Press.
- (Czarnecki & Eisenecker, 1999) Czarnecki, K., Eisenecker, U. (1999). *Generative Programming: Methods, Techniques and Applications*. Addison-Wesley.
- (Czarnecki, 2000) Czarnecki, K. (2000). *Generative Programming: Methods, tools and Applications* (2nd edition). Addison-Wesley.

- (Dahn, 2002) Dahn, I. (2002). Slicing Books - The Author's Perspective. In R. Bromme, E. Stahl (Eds.), *Writing Hypertext and Learning*, Pergamon (pp. 125-151).
- (Dalziel, 2003) Dalziel, J. (2003). Implementing Learning Design: The Learning Activity Management System (LAMS). Actas del congreso *20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE 2003)*, Adelaide, Australia.
- (Dalziel, 2005) Dalziel, J. (2005). *From re-usable e-learning content to re-usable learning designs: Lessons from LAMS*. Technical report, LAMS Foundation.
- (Dalziel, 2006) Dalziel, J. (2006), Lessons from LAMS for IMS Learning Design. Actas del congreso *IEEE International Conference on Advanced Learning Technologies (ICALT'06)* (pp.1101-1102).
- (Deursen, 1997) Deursen, A. (1997). Domain-Specific Languages versus Object-oriented Frameworks: A Financial Engineering case Study. Actas del congreso *Smalltalk and Java in Industry and Academia, STJA '97*. Ilmenau Technical University.
- (Deursen & Klint, 1998) Deursen, A. & Klint, P. (1998). Little Languages: Little maintenance ?. *Journal of Software Maintenance*, 10, 75-92.
- (Deursen et al., 2000) Deursen, A., Klint, P. & Visser, J. (Junio, 2000). Domain-Specific Languages: An Annotated Bibliography. *ACM SIGPLAN Notices*. 35(6), 26-36.
- (Dick et al., 2000) Dick, W., Carey, L. & Carey, J.O. (2000). *The Systematic Design of Instruction* (5th edition). Allyn & Bacon.
- (Dodero et al., 2006) Dodero, J. M., Tattersall, C., Burgos, D., & Koper, R. (2006). Non-representational authoring of learning designs: from idioms to model-driven development. Disponible online en: <http://hdl.handle.net/1820/783> (último acceso, Mayo 2008).
- (Dodero et al., 2007) Dodero, J.M., Tattersall, C., Burgos, D. & Koper, R. (2007). Transformational Techniques for Model-Driven Authoring of Learning Designs. Actas del congreso *6th International Conference in Web Based Learning (ICWL 2007)* (pp. 230-241).
- (Dodero et al., 2010) Dodero, J.M., Martínez del Valle, A. & Torres, J. (Diciembre, 2010). An extensible approach to visually editing adaptive learning activities and designs based on services. *Journal of Visual Languages & Computing*, 21(6).
- (Dolan et al., 2005) Dolan, D., Delgado-Kloos, C., Muñoz-Organero, M. & Pardo, A. (2005). E-LANE: Open Source eLearning in Latin America. Actas del congreso *Open Source for Education in Europe, Research & Practice Conference* (pp. 117-126), Heerlen, Países Bajos.
- (Dougherty, 1999) Dougherty, D. (1999). The Making of DocBook DTD. *O'Reilly XML.com*. Disponible online en: <http://www.xml.com/pub/a/1999/10/docbook/docbook-making.html>.

- (Downes, 2001) Downes, S. (2001). Learning Objects: Resources For Distance Education Worldwide. *The International Review of Research in Open and Distance Learning*, 2(1).
- (Dumas et al., 2005) Dumas, M., Aalst, W. M. P. & Hofstede, A. H. t. (2005). *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley-Interscience.
- (Durand & Downes, 2009) Durand, G.; Downes, S. (Julio, 2009). Toward Simple Learning Design 2.0. Actas del congreso *4th International Conference on Computer Science & Education (ICCSE '09)* (pp. 894-897).
- (Durand et al., 2010) Durand, G., Belliveau, L. & Craig, B. (2010). Simple Learning Design 2.0. Actas del congreso *IEEE 10th International Conference on Advanced Learning Technologies (ICALT'10)* (pp.549-551).
- (Durm et al., 2001) van Durm, R., Duval, E., Verhoeven, B., Cardinaels, K., & Oliivié, H. (2001). Extending the ARIADNE web-based learning environment. Actas del congreso *World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2001)* (pp. 1932–1937), Tampere, Finland.
- (Duval et al., 2001) Duval, E., Forte, E., Cardinaels, K., Verhoeven, B., Van Durm, R., Hendrikx, K., Forte, M. W., Ebel, N., Macowicz, M., Warkentyne, K., & Haenni, F. (Mayo, 2001). The ARIADNE knowledge pool system. *Communications of the ACM*, 44(5), 72-78.
- (Emerson, 1990) Emerson, E.A. (1990). Temporal and Modal Logic. In J.V. Leeuwen (Ed.) *Handbook of Theoretical Computer Science Vol. B: Formal Models and Semantics*, (pp. 995-1072). Massachusetts: MIT Press.
- (Escobedo et al., 2007) Escobedo del Cid, J. P., de la Fuente Valentín, L., Gutiérrez, S., Pardo, A. & Delgado Kloos, C. (2007). Implementation of a Learning Design Run-Time Environment for the .LRN Learning Management System. *Journal of Interactive Media in Education*.
- (Farrel et al., 2004) Farrell, R., Liburd, S. & Thomas, J. (2004). Dynamic Assembly of Learning Objects. Actas del congreso *13th international World Wide Web conference on Alternate track papers & posters 2004* (pp. 162-169).
- (Fernández-Manjón and Fernández-Valmayor, 1997) Fernández-Manjón, B., Fernández-Valmayor, A. (Septiembre, 1997). Improving World Wide Web Educational Uses Promoting Hypertext and Standard General Markup Languages. *Education and Information Technologies*, 2(3), 193-206.
- (Fernandez-Manjon and Sancho, 2002) Fernandez-Manjon, B. & Sancho, P. (2002). Creating cost-effective adaptive educational hypermedia based on markup technologies and e-learning standards. *Interactive Educational Multimedia*, 4, 1-11.

- (Fernández-Manjón et. al., 2007) Fernández-Manjón, B., Sierra, J.L., Moreno-Ger, P., Martínez-Ortiz, I. *Uso de estándares aplicados a TIC en Educación*. Informe 16 del Centro Nacional de Información y Comunicación Educativa. Ministerio de Educación, Política Social y Deporte . Subdirección General de Información y Publicaciones. NIPO 651-06-344-7, ISBN 10: 84-369-4316-3. Disponible online en <http://ares.cnice.mec.es/informes/16/contenido/indice.htm> (último acceso Junio 2010).
- (Fernández-Manjón et al., 2009) Fernández-Manjón, B., Sierra, J.L., Martínez-Ortiz, I., Moreno-Ger, P. *Estandarización y Modelado Educativo*. Informe nº 20. Instituto Superior de Formación y Recursos en Red para el Profesorado (ISFTIC) (anteriormente conocido como CNICE, Centro Nacional de Información y Comunicación Educativa) del Ministerio de Educación, Política Social y Deporte (MEPSYD) de España. ISBN: 978-84-369-4780-9 (en prensa) . 2009
- (Ferraris et al., 2008) Ferraris, C., Martel, C. & Vignollet, L. (2008). Modelling the “Planet Game” Case Study with LDL and Implementing it with LDI. *Journal of Interactive Media in Education*, 2008/12. Disponible online en: <http://jime.open.ac.uk/article/2008-20/361> (última visita: 15 de Diciembre de 2010).
- (Ferraris et al., 2008b) Ferraris, C., Martel, C. & Vignollet, L. (2008). LDL for Collaborative Activities. In. Luca Botturi & Todd Stubbs (Eds.) *Handbook of Visual Languages in Instructional Design: Theories and Practices* (pp. 226-253). Hershey, PA: Idea Group.
- (Ferrucci et. al., 2002) Ferrucci, F., Tortora, G. & Vitiello, G., (2002). Exploiting Visual Languages in Software Engineering. In *Handbook of Software Engineering & Knowledge Engineering* (pp. 53-76).
- (Fisler et al., 2005) Fisler, J., Bleisch, S. & Niederhuber, M. (Junio, 2005). Development of sustainable e-learning content with the open source eLesson Markup Language eLML. Actas del congreso *ISPRS Workshop*, Potsdam, Germany.
- (Fowler, 2010) Fowler, M. (2010). *Domain Specific Languages*. Addison-Wesley.
- (Frankel, 2003) Frankel, D. S. (2003). *Model Driven Architecture: Applying MDA to Enterprise Computing*. Wiley, Ed.
- (Friedman et al., 2001) Friedman, D., Wand, M. & Hayes, C.T. (2001). *Essentials of Programming Languages* (2nd edition). Massachusetts: MIT Press.
- (Friesen, 2005) Friesen N. (2005). Interoperability and Learning Objects: An Overview of e-Learning Standardization. *Interdisciplinary Journal of Knowledge and Learning Objects*, 2005(1), 23-31. Disponible online en: <http://ijklo.org/Volume1/v1p023-031Friesen.pdf> (último acceso Enero de 2009).

- (Fuentes-Fernández et al., 2007) Fuentes-Fernández, R., Gómez-Sanz, J. & Pavón, J. (2007). Managing Contradictions in Multi-Agent Systems. *IEICE Transactions on Information and Systems*. E90-D(8), 1243-1250.
- (Gamma et al., 1994) Gamma, E., Helm, R., Johnson R. & Vlissides, J. M. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley.
- (Gibson, 2003) Gibson, F. P. (2003). Supporting Learning in Evolving Dynamic Environments. *Computational & Mathematical Organization Theory*, 9(4), 305-326.
- (Goldfarb, 1981) Goldfarb, C.F., 1981, A Generalized Approach to Document Markup. Actas del congreso *ACM SIGPLAN SIGOA Symposium on Text manipulation* (pp. 68-73).
- (Gómez-Pérez, A., et al., 2004) Gómez-Pérez, A., Fernández-López, M. & Corcho, O., *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. Berlin Heidelberg: Springer.
- (González-Barahona et al., 2005) Gonzalez-Barahona, J.M., Dimitrova, V., Chaparro, D., Tebb, C. & Mazza, R. (2005). Producing educational resources in the libre way: The Edukalibre project. Actas del congreso *IADIS International Conference WWW/Internet*, Lisboa, Portugal.
- (González-Barahona et al., 2005b) Gonzalez-Barahona, J.M., Tebb, C., Dimitrova, V., Chaparro, D. & Romera, T. (2005). Transferring Libre Software Development Practices to the Production of Educational Resources: the Edukalibre Project. Actas del congreso *First International Conference on Open Source Systems*.
- (González-Barbone & Anido-Rifón, 2008) Gonzalez-Barbone, V. & Anido-Rifón, L. (Diciembre, 2008). Creating the first SCORM object. *Computers & Education*, 51(4), 1634-1647.
- (Goodwill & Hightower, 2004) Goodwill, J. & Hightower, R. (2004). *Professional Jakarta Struts*. Indianapolis: Wiley.
- (Griffiths et al., 2005) Griffiths, D., Blat, J., Garcia, R., Vogten, H., & Kwong, K.L. (2005). Learning Design Tools. In R. Koper, C. Tattersall (Eds.) *Learning Design - A Handbook on modelling and Delivering Networked Education and Training* (pp. 109-135). Heidelberg: Springer.
- (Griffiths et al., 2008) Griffiths, D., Beauvoir, P. & Sharples, P. (2008). Advances in Editors for IMS-LD in the TENCompetence Project. Actal del congreso *8th IEEE International Conference on Advanced Learning Technologies (ICALT'08)* (pp. 1045-1047).
- (Grigoreanu et al., 2009) Grigoreanu, V., Fernandez, R., Inkpen, K. & Robertson, G. (2009). What designers want: Needs of interactive application designers. Actas del congreso *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'09)* (pp. 139-146).
- (Gronback, 2009) Gronback, R. C. (2009). *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit*. Addison-Wesley Professional.

- (Gueye et al., 2004) Gueye, B., Rigaux, P. & Spyrtos, N. (2004). Taxonomy-Based Annotation of XML Documents: Application to eLearning Resources. Actas del congreso *Methods and Applications of Artificial Intelligence: Third Hellenic Conference on AI* (SETN 2004) (pp. 33-42).
- (Herndon & Berzins, 1988) Herndon, R.M. & Berzins, V.A. (1988). The Realizable Benefits of a Language Prototyping Language. *IEEE Transactions on Software Engineering*, 14, 803–809.
- (Hernández-Leo et. al., 2006) Hernández-Leo, D., Harrer, A., Dodero, J. M., Asension-Pérez, J. I., & Burgos, D. (2006). Creating by reusing Learning Design solutions. Actas del congreso *8th Simposio Internacional de Informática Educativa*. Disponible online en: <http://dspace.learningnetworks.org>.
- (Hernández et. al., 2007) Hernández, R., Pardo, A. & Delgado Kloos, C. (2007). Creating and deploying effective elearning experiences using .LRN. *IEEE Transactions on Education*, 50(4), 345-351.
- (Heyer et. al., 2007) Heyer, S., Oberhuemer, P., Zander, S. & Prenner, P. (2007). Making Sense of IMS Learning Design Level B: From Specification to Intuitive Modeling Software. In E. Duval, R. Klamma, M. Wolpers (Eds.), *Creating New Learning Experiences on a Global Scale*. 2nd European Conference on Technology Enhanced Learning (EC TEL 2007) (pp. 86-100). Berlin: Springer.
- (Hudak, 1998) Hudak, P. (1988). Domain-Specific Languages. In *Handbook of Programming Languages V. III: Little Languages. and Tools*. Macmillan Tech. Publishing.
- (Ibrahim, 1989) Ibrahim, B. (1989). Software Engineering Techniques for CAL. *Computers & Education*, 5, 215-222.
- (IEEE, 2002) *IEEE Standard for Learning Object Metadata*. (2002).
- (IEEE, 2006) *Institute of Electrical and Electronics Engineers (IEEE) Learning Technology Standards Committee (IEEE-LTSC)*. Disponible online en: from <http://ltsc.ieee.org/>.
- (IMS, 2002) IMS-QTI-ASI_INFO (2002). *IMS Question & Test Interoperability: ASI Information Model Specification*, IMS Global Learning Consortium.
- (IMS, 2003) IMS-LD (2003). *IMS Learning Design Information Model Version 1.0 Final Specification*. IMS Global Learning Consortium.
- (IMS, 2003b) IMS-SS (2003). *IMS Content Simple Sequencing Information and Behavior Model Version 1.0 Final Specification*, IMS Global Learning Consortium.
- (IMS, 2004) IMS. *IMS Content Packaging Information Model Version 1.1.4 Final Specification*. Disponible online en: <http://www.imsglobal.org/content/packaging/> (último acceso Junio 2009).

- (IMS, 2005) IMS. *IMS Learner Information Package Summary of Changes Version 1.0.1 Final Specification*. (2005).
- (IMS, 2006) *IMS Global Learning Consortium*. (2006). Disponible online en: <http://www.imsglobal.org>.
- (IMS, 2006b) IMS_QTI2-INFO (2006). *IMS QTI and Test Interoperability Assessment Test, Section, and Item Information Model*, IMS Global Learning Consortium.
- (IMS, 2006c) IMS (2006). *IMS Question and Test Interoperability Assessment Test, Section, and Item Information Model Version 2.1 Public Draft revision 2 Specification*. Disponible online en: <http://www.imsglobal.org/question/>.
- (Johnson, 1975) Johnson, S.C. (1975) YACC-yet Another Compiler-Compiler. *Computing Science Technical Report 32*. AT&T Bell Laboratories.
- (Karampiperis & Sampson, 2004) Karampiperis, P. & Sampson, D. (2004). A flexible authoring tool supporting adaptive learning activities. Actas del congreso *IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2004)*.
- (Karampiperis & Sampson, 2007) Karampiperis, P. & Sampson, D. (Julio, 2007). Towards a Common Graphical Language for Learning Flows: Transforming BPEL to IMS Learning Design Level A Representations. Actas del congreso *7th IEEE International Conference on Advanced Learning Technologies (ICALT'07)* (pp.798-800).
- (Kernighan, 1981) Kernighan, B.W. (1981). *PIC: A Crude Graphics Language for Typesetting. Technical Report*. Bell Laboratories.
- (Kiebertz et al., 1996) Kiebertz, R. B., McKinney, L., Bell, J. M., Hook, J., Kotov, A., Lewis, J., Oliva, D. P., Sheard, T., Smith, I. & Walton, L. (1996). A Software Engineering Experiment in Software Component Generation. Actas del congreso *18th International Conference on Software Engineering ICSE-18*.
- (Kim et al. 2005) Kim, K.-H., Yoo, H.-J. & Kim, H.-S. (2005). A process-driven e-learning content organization model. Actas del congreso *ICIS 2005* (pp. 328-333). IEEE Computer Society.
- (Kleppe, 2008) Kleppe, A. (2008) *Software Language Engineering: Creating Domain-Specific Languages Using Metamodels*. Addison-Wesley Professional.
- (Klint et al., 2005) Klint, P., Lämmel, R. & Verhoef, C. (Julio, 2005). Toward an Engineering Discipline for Grammarware. *ACM Transactions on Software Engineering and Methodology*, 14(3), 331–380.
- (Knuth, 1984) Knuth, D. E. (1984). Literate programming. *Computer Journal*, 27(1), 97-111.
- (Knuth, 1984b) Knuth, D.E. (1984). *The TEXBook*. Addison-Wesley.

- (Koch, 2002) Koch, M. (2002). Interoperable Community Platforms and Identity Management in the University Domain. *International Journal on Media Management*, 4(1), 21-30.
- (Koper, 2000) Koper, R. (2000). *From Change to Renewal: Educational Technology Foundations of Electronic Learning Environments*. Open University of the Netherlands.
- (Koper, 2001) Koper, R. (2001). *Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML*. Educational Technology Expertise Centre (OTEC), Open University of the Netherlands.
- (Koper, 2003) Koper, R. (2003). Combining re-usable learning resources and services to pedagogical purposeful units of learning. In A. Littlejohn, *Reusing Online Resources: A Sustainable Approach to eLearning* (pp. 46-59).
- (Koper & Burgos, 2005) 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.
- (Koper & Manderveld, 2004) Koper, E. J. R., & Manderveld, J. M. (2004). Educational modelling language: modelling reusable, interoperable, rich and personalised units of learning, *British Journal of Educational Technology*, 35 (5), 537-552.
- (Koper & Olivier, 2004) Koper, R. & Olivier, B. (2004). Representing the Learning Design of Units of Learning, *Educational Technology & Society*, 7(3), 97-111.
- (Koper & Tattersall, 2005) Koper, R. & Tattersall, C. (Eds.) (2005). *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training*. Heidelberg: Springer Verlag.
- (Krasner & Pope, 1988) Krasner, G.E., Pope, T.S. (Agosto/Septiembre, 1988). A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk 80 System. *Journal of Object Oriented Programming*. 1(3), 26-49.
- (Krueger, 1992) Krueger, C.W. (1992). Software Reuse. *ACM Computing Surveys*, 24(2), 131-183.
- (Laforcade, 2005) Laforcade, P (2005). Towards a UML-based educational modeling language. Actas del congreso *Fifth IEEE Int. Conf. on Advanced Learning Technologies (ICALT'05)* (pp 855-859).
- (Laforcade, 2007) Laforcade, P. (2007). Graphical Representation of Abstract Learning Scenarios: the UML4LD Experimentation. Actas del congreso *7th IEEE International Conference on Advanced Learning Technologies (ICALT 2007)* (pp. 477-479).
- (LAMS, 2005) LAMS. (2005). LAMS v 1.0.2. Disponible online en: <http://www.lamsfoundation.org/>.

- (Le Pallec et al. 2006) Le Pallec, X., Filho, C.Od.M., Marvie, R., Nebut, M. & Tarby, J.-C. (2006). Supporting Generic Methodologies to Assist IMS-LD Modeling. Actas del congreso 6th *IEEE International Conference on Advanced Learning Technologies (ICALT'06)* (pp.923-927).
- (Li, 1991) Li, X. (1991). What's So Bad About Rule-Based Programming?. *IEEE Software* 8, 103-105.
- (Makoto et al., 2005) Makoto, M., Lee, D., Mani, M. & Kawaguchi, K. (Noviembre, 2005). Taxonomy of XML schema languages using formal language theory. *ACM Transactions on Internet Technology*, 5(4), 660-704.
- (Marriott et al., 1999) Marriott, K., Meyer, B., Wittenburg, K.B.A. (1999). Survey of Visual Language Specification and Recognition. In Marriott K, Meyer B, (Eds.) *Visual Language Theory* (pp. 5-85). Berlin: Springer.
- (Martel et al., 2006) Martel, C., Vignollet, L., Ferraris, C., David, J.P. & Lejeune. A. (2006). LDL: An Alternative EML. Actas del congreso 6th *IEEE International Conference on Advanced Learning Technologies (ICALT'06)* (pp. 1107-1108).
- (Martel et al., 2006b) Martel, C., Vignollet, L., Ferraris, C. & Durand, G. (2006). LDL: a Language to Model Collaborative Learning Activities. In E. Pearson & P. Bohman (Eds.), Actas del congreso *World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006* (pp. 838-844). Chesapeake, VA: AACE.
- (Martens & Vogten, 2005) Martens, H. & H. Vogten (2005). Reference Implementation of a Learning Design Engine. In R. Koper and C. Tattersall (Eds.) *Learning Desing: a Handbook on Modelling and Delivering Networked Education and Training* (pp. 91-108). Heidelberg: Springer.
- (Martinez-Ortiz et al., 2005) Martinez-Ortiz, I., P. Moreno-Ger, Sancho-Thomas, P., Fernández-Manjón, B. (2005). Using DocBook To Aid in the Creation of Learning Content. In A. Rettberg & C. Bobda (Eds.) *New Trends and Technologies in Computer-Aided Learning for Computer-Aided Design*. Heidelberg: Springer.
- (Martinez-Ortiz et al., 2006) Martinez-Ortiz, I., P. Moreno-Ger, Sierra, J.L., Fernández-Manjón, B. (2006). Production and Deployment of Educational Videogames as Assessable Learning Objects. Actas del congreso *First European Conference on Technology Enhanced Learning (ECTEL 2006)*.
- (Martínez-Ortiz et al. 2006b) Martínez-Ortiz, I., Moreno-Ger, P., Sierra, J.L., Fernández-Manjón, B. (2006), A General Architecture for the Authoring and the Operationalization of e-Learning Applications with Educational Modelling Languages. Actas del congreso *Simposio Internacional de Informática Educativa (SIIIE06)*, (pp. 280-288).

- (Martínez-Ortiz et al., 2006c) Martínez-Ortiz I, Moreno-Ger P, Sierra JL, Fernández-Manjón B. (Junio, 2006). Using DocBook and XML Technologies to Create Adaptive Learning Content. *International Journal of Computer Science and Applications*, 3(2), 91-108.
- (Martínez-Ortiz et al. 2007) Martínez-Ortiz, I., Moreno-Ger, P., Sierra, J.L., Fernández-Manjón, B. (2007). Educational Modeling Languages: A Conceptual Introduction and a High-Level Classification. In Fernández-Manjón, B, Sánchez, J.M et al. (Eds.) *Computers and Education: E-learning – from theory to practice* (pp. 27-40). Heidelberg: Springer.
- (Martínez-Ortiz et al., 2007b) Martínez-Ortiz, I., Moreno-Ger, P., Sierra, J.L., Fernández-Manjón, B. (2007). Supporting the Authoring and Operationalization of Educational Modelling Languages, *Journal of Universal Computer Science*, 13(7), 938-947.
- (Martínez-Ortiz et al., 2008) Martínez-Ortiz, I., Moreno-Ger, P., Sierra, J.L., Fernández-Manjón, B. (2008). A Flow-Oriented Visual Language for Learning Designs. In Li F, Zhao J, Shih TK, Lau R, Li Q, McLeod D, (Eds.) *Advances in Web Based Learning - ICWL 2008. 7th International Conference on Web-based Learning (ICWL 2008)* (pp. 486-496). Berlin: Springer.
- (Martínez-Ortiz et al., 2008b) Martínez-Ortiz, I., Sierra, J.L., Fernández-Manjón, B. (2008). Enhancing Reusability of IMS-LD Units of Learning: The e-LD Approach. Actas del congreso *8th IEEE International Conference on Advanced Learning Technologies (ICALT 2008)* (402-404). Washington DC, EEUU: IEEE Computer Society.
- (Martínez-Ortiz et al., 2009) Martínez-Ortiz, I., Sierra, J.L., Fernández-Manjón, B. (Abril, 2009) Translating e-learning Flow-Oriented Activity Sequencing Descriptions into Rule-based Designs. Actas del congreso *International Conference on Information Technology: New Generations (ITNG 2009)* (pp. 1108-1113). Las Vegas, EEUU. Washington DC, EEUU: IEEE Computer Society.
- (Martínez-Ortiz et al., 2009b) Martínez-Ortiz, I., Sierra, J.L., Fernández-Manjón, B., Fernández-Valmayor, A., (Septiembre, 2009). Language engineering techniques for the development of e-learning applications. *Journal of Network and Computer Applications*, 32(5): 1092-1105.
- (Martínez-Ortiz et al., 2009c) Martínez-Ortiz, I., Sierra, J.L., Fernández-Manjón, B. (Marzo, 2009). Authoring and Reengineering of IMS Learning Design Units of Learning. *IEEE Transactions on Learning Technologies*, 2(3), 189-202.
- (Martínez-Ortiz et al., 2009d) Martínez-Ortiz I, Sierra JL, Fernández-Manjón B. Enhancing IMS-LD Units of Learning Comprehension. Actas del congreso *4th International Conference on Internet and Web Applications and Services (ICIW 2009)* (pp. 561-566).
- (Mayes & de Freitas, 2005) Mayes, T., de Freitas, S. (2004). *Stage 2: Review of e-learning theories, frameworks and models*. JICS E-learning Model Desk Study. Disponible en:

- [http://www.jisc.ac.uk/uploaded_documents/Stage%20%20Learning%20Models%20\(Versión%201\).pdf](http://www.jisc.ac.uk/uploaded_documents/Stage%20%20Learning%20Models%20(Versión%201).pdf) (último acceso: Octubre 2009).
- (Mayrhauser & Vans, 1995) Mayrhauser, A. v. & Vans, A. M (1995). Program understanding: models and experiments, *Advances in Computers*, 40, 1-38.
- (McAndrew et al., 2004) McAndrew, P., Woods, W.I.S., Little, A., Weller, M.J., Koper, R., & Vogten, H. (2004). Implementing learning design to support web-based learning. Actas del congreso *Australasian World Wide Web Conference (AusWeb04)*.
- (McMartin, 2004) McMartin, F. (Septiembre, 2004). MERLOT: A Model for User Involvement in Digital Library Design and Implementation, *Journal of Digital Information*, 5(3).
- (Menon & Pingali, 1999) Menon, V. & Pingali, K. (Octubre, 1999). A Case for Source-Level Transformations in MATLAB. Actas del congreso *Second USENIX Conference on Domain-Specific Languages*. USENIX Association.
- (Mernik et al., 2005) Mernik, M., Heering, J., Sloane, A.M. (Diciembre, 2005). When and how to Develop Domain-Specific Languages. *ACM Computing Surveys*. 37(4), 316-344.
- (Merrill, 1994) Merrill, M. D. (1994). *Instructional Design Theory*. Englewood Cliffs.
- (Miao, 2005) Miao, Y. (2005). Facilitating learning designers to author units of learning using IMS-LD. Actas del congreso *International Conference on Computers in Education* (pp. 275-282).
- (Milligan et al., 2005) Milligan, C.D., Beauvoir, P &, Sharples, P. (2005). The Reload Learning Design Tools, *Journal of Interactive Media in Education*, 2005(7).
- (Molenda, 2003) Molenda, M. (Mayo/Junio, 2003). In Search of the Elusive ADDIE Model. *Performance Improvement Journal*, 42(5), 34-36.
- (Molina et. al., 2008) Molina, A.I., Jurado, F., Giraldo, W.J., Redondo, M.A. & Ortega, M. (Julio, 2008). Specifying Scripts and Collaborative Tasks in CSCL Environment Using IMS-LD and CIAN. Actas del congreso 8th *IEEE International Conference on Advanced Learning Technologies (ICALT '08)* (pp. 775-777).
- (Molloy, 2003) Molloy, D. (2003). Single Source Interactive and Printed Content Publishing Using the DocBook XML Standard. Actas del congreso *Second International Conference on Multimedia and ICTs in Education m-ICTE*.
- (Moreno-Ger et al., 2005) Moreno-Ger, P., Martínez-Ortiz, I., Fernández-Manjón, B. (2005). The <e-Game> Project: Facilitating the Development of Educational Adventure Games. Actas del congreso *International Conference on Cognition and Exploratory Learning in the Digital Age (CELDA 2005)*.
- (Moreno-Ger et al., 2006) Moreno-Ger, P., Martínez-Ortiz, I., Sierra, J.L., Fernández-Manjón, B. (Julio, 2006). A Descriptive Markup Approach to Facilitate the Production of e Learning

- Contents. Actas del congreso *6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006)* (pp. 19-21). Washington DC, USA: IEEE Computer Society.
- (Moreno-Ger et al., 2007) Moreno-Ger, P., Sierra, J.L., Martínez-Ortiz, I., Fernández-Manjón, B. (Junio, 2007). A Documental Approach to Adventure Game Development. *Science of Computer Programming*, 67(1), 3-31.
- (Moreno-Ger et al., 2008) Moreno-Ger, P., Sierra, J.L., Martínez-Ortiz, I., Fernández-Manjón, B. (2008). A Content-Centric Development Process Model. *IEEE Computer*, 41 (3), 24-30.
- (Moreno-Ger et al., 2009) Moreno-Ger, P., Fuentes-Fernández, R., Sierra, J.L., Fernández-Manjón, B. (Marzo, 2009). Model-checking for Adventure Videogames. *Information and Software Technology*, 51(3), 564-580.
- (Mosses, 2004) Mosses, P.D. (Julio-Diciembre, 2004). Modular structural operational semantics. *Journal of Logic and Algebraic Programming*, 60-61, 195-228.
- (Mosses, 2006) Mosses, P.D. (Febrero, 2006). Formal Semantics of Programming Languages: An Overview. *Electronic Notes in Theoretical Computer Science*, 148(1), 41-73.
- (Muñoz Merino et. al., 2009) Muñoz Merino, P.J., Delgado Kloos, C., Fernández Naranjo, J. (2009). Enabling Interoperability for LMS Educational Services. *Computer Standards & Interfaces*, 31(2), 484-498.
- (Murata, 1989) Murata, T. (Abril, 1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4), 541-580.
- (Neighbors, 1986) Neighbors, J. (1986). The Draco approach to constructing software from reusable components. In *Readings in artificial intelligence and software engineering* (pp. 525-535). Morgan-Kaufmann.
- (Neumann et. al., 2009) Neumann, S., Klebl, M., Griffiths, D., Hernandez-Leo, D., De la Fuente-Valentin, L., Hummel, H. G. K., Brouns, F., Derntl, M., & Oberhuemer, P. (in press). Report of the Results of an IMS LEarning Design Expert Workshop. *International Journal of Emerging Technologies (iJET)*.
- (OMG 2005) Object Management Group. (2005). *Unified Modelling Language version 2.0: Superstructure*.
- (OMG, 2005b) Object Management Group. (2005). *Unified Modelling Language version 2.0: Infrastructure*.
- (OMG, 2005c). Object Management Group. (2005). *MOF 2.0/XMI Mapping Specification, v2.1 (No. formal/05-09-01)*.
- (OMG, 2010) Object Management Group. (2010). *Business Process Model and Notation version 2.0. dtc/2010-06-05.2010*. Disponible online en: <http://www.omg.org/spec/BPMN/2.0> (último acceso: Noviembre de 2010).

- (OUNL, 2005) OUNL. (2005). *Coppercore v3.0*. Disponible online en: <http://www.coppercore.org>.
- (Ouyang et. al., 2006) Ouyang, C., Aalst, W.M.P. van der, Dumas, M. & Hofstede, A.H.M. ter. (2008). *Translating BPMN to BPEL*. BPM Center Report BPM-06-02. Disponible online en: <http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/reports/2006/BPM-06-02.pdf> (último acceso Noviembre de 2008).
- (Paakki, 1995) Paakki J. (1995). Attribute Grammar Paradigms – A High-Level Methodology in Language Implementation. *ACM Computing Surveys*, 27(2), 196-255.
- (Papzoglou & Georgakopoulos, 2003) Papzoglou, M.P. & Georgakopoulos, D. (2003). Service-Oriented Computing. *ACM Communications*, 46(10), 24-28.
- (Paquette et al., 1997) Paquette, G., Crevier, F. et al. (1997). Méthode d'ingénierie d'un système d'apprentissage (MISA). *REview Informations In Cognito*, 8.
- (Paquette, 2001) Paquette, G. (2001). TeleLearning Systems Engineering - Towards a new ISD model. *Journal of Structural Learning* 14, 1-35.
- (Paquette, 2004) Paquette, G. (2004). Educational Modeling Languages, from an Instructional Engineering Perspective. In R. McGreal *Online education using learning objects* (pp. 331-246). London: Routledge/Falmer.
- (Paquette et al., 2005) Paquette, G., I. de la Teja, Léonard, M., Lundgren-Cayrol, K. & Marino, O. (2005). Using an Instructional Engineering Method and a Modeling Tool. In R. Koper and C. Tattersall *Learning Desing: Modelling and Implementing Network-based Education & Training*. Heidelberg: Springer-Verlag.
- (Paquette et al., 2005b) Paquette, G., Marino, O., De la Teja, I., Lundgren-Cayrol, K., Léonard, M. & Contamines, J. (2005). Implementation and Deployment of the IMS Learning Design Specification. *Canadian Journal of Learning and Technology*, 31(2).
- (Paquette et al., 2006) Paquette, G., Léonard, M., Lundgren-Cayrol, K., Mihaila, S. & Gareau, D. (2006). Learning Design based on Graphical Knowledge-Modeling. *Educational Technology & Society*, 9(1), 97-112.
- (Parr, 2007) Parr T. (2007). *The Definitive ANTLR Reference: Building Domain-Specific Languages*. Pragmatic Bookshelf.
- (Peltz, 2003) Peltz, C. (Octubre, 2003). Web services orchestration and choreography. *IEEE Computer*, 36(10), 46-52.
- (Pereira & Warren, 1980) Pereira, F.C.N., Warren, D.H.D. (1980). Definite Clause Grammars for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks. *Artificial Intelligence*, 13(3), 231-278.

- (Plotkin, 2004) Plotkin, G.D. (Julio-Diciembre, 2004). A Structural Approach to Operational Semantics. *Journal of Logic and Algebraic Programming*, 60-61, 17-139.
- (Polsani, 2003) Polsani, P. (2003). Use and Abuse of Reusable Learning Objects. *Journal of Digital Information*, 3(4).
- (Rathz et al., 2004) Rathz, S., Walsh, N. and Burnard, L., 2004, A unified model for text markup: TEI, Docbook, and beyond (Proceedings of XML Europe 2004).
- (Ramaswamy et al., 1997) Ramaswamy, M., Sarkar, S., Chen, Y.S (1997). Using Directed Hypergraphs to Verify Rule-Based Expert Systems. *IEEE Transactions on Knowledge and Data Engineering* 9(2). 221-237.
- (Rawlings et al., 2002) Rawlings, A., Rosmalen, P. v., Koper, R., Artacho-Rodríguez, M., Lefrere, P. (2002). *Survey of Educational Modelling Languages (EML)*, CEN/ISSS WS/LT Learning Technologies Workshop.
- (Recker & Mendling, 2006) Recker, J. & Mendling J. (2006). On the Translation between BPMN and BPEL: Conceptual Mismatch between Process Modeling Languages. *Actas del congreso 18th International Conference on Advanced Information Systems Engineering* (pp. 521-532).
- (Reigeluth, 1983) Reigeluth, C. M. (1983). *Instructional Design Theories and Models: An Overview of Their Current Status*. Lawrence Erlbaum Associates.
- (Rodríguez-Artacho et al., 1999) Rodríguez-Artacho, M., Verdejo, M. E., Mayorga, J.J. & Calero, M.Y. (1999). Using High-Level Language to Describe and Create Web-Based Learning Scenarios. *Actas del congreso IEEE Frontiers In Education FIE '99*.
- (Salus, 1998) Salus, P.H. (Ed). (1998). *Handbook of Programming Languages, Volume III. Little Languages and Tools*. Macmillan Tech. Publishing.
- (Sarasa-Cabezuelo et al., 2008) Sarasa-Cabezuelo, A., Navarro-Iborra, A., Sierra, J.L. & Fernández-Valmayor, A. (2008). Building a Syntax Directed Processing Environment for XML Documents by Combining SAX and JavaCC. *Actas del congreso 19th International Workshop on Database and Expert Systems Applications (DEXA 2008)* (pp. 256-260). Washington DC, USA: IEEE Computer Society.
- (Schelnoff et al., 2000) Schlenoff, C., Gruninger M., Tissot, F., Valois, J., Lubell, J., & Lee, J. (2000). *The Process Specification Language (PSL): Overview and Version 1.0 Specification*. NISTIR 6459. Gaithersburg, MD: National Institute of Standards and Technology.
- (Sethi, 1996) Sethi, R. (1996). *Programming Languages: Concepts and Construct*. Boston, MA: Addison Wesley.
- (Short et al., 2004) Short, K., Cook, S. & Greenfield, K. J. (2004). *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*. John Wiley and Sons, Inc.

- (Sierra et al., 2004) 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.
- (Sierra et al., 2005) Sierra, J. L., Fernández-Manjón, B., Fernández-Valmayor, A. & Navarro, A. (2005). Document-Oriented Construction of Content-Intensive Applications, *International Journal of Software Engineering and Knowledge Engineering*, 15(6), 975-993.
- (Sierra et al., 2005b) Sierra-Rodriguez, J. L., Martínez-Ortiz, I., Moreno-Ger, P., Lopez-Moratalla, J., Fernández-Manjón, B. (2005). Building Learning Management Systems Using IMS Standards: Architecture of a Manifest Driven Approach. In *Lecture Notes in Computer Science 3583* (pp. 144-156.).
- (Sierra et al., 2005c) Sierra-Rodriguez, J. L., Navarro, A., Fernández-Valmayor, A. & Fernández-Manjón, B. (2005). Incremental Definition and Operationalization of Domain-Specific Markup Languages in ADDS, *ACM SIGPLAN Notices*, 40(12), 28-37.
- (Sierra, et al., 2006) Sierra-Rodriguez, J. L., Fernández-Valmayor, A. & Fernández-Manjón, B. (2006). A Document Oriented Paradigm for the Construction of Content-Intensive Applications, *The Computer Journal*, 49(5), 562-584.
- (Sierra et al., 2006b) Sierra, J.L., Fernández-Valmayor, A., Guinea, M. (2006). Exploiting Author-Designed Domain-Specific Descriptive Markup Languages in the Production of Learning Content. Actas del congreso *6th International Conference on Advanced Learning Technologies (ICALT 2006)* (pp. 519-524). Washington DC, EEUU: IEEE Computer Society.
- (Sierra et al., 2006c) Sierra, J.L., Fernández-Valmayor, A., Guinea, M., Hernánz, H. (2006). From Research Resources to Virtual Objects: Process model and Virtualization Experiences. *Journal of Educational Technology & Society*, 9(3), 56-68.
- (Sierra & Fernández-Valmayor, 2007) Sierra, J.L., Fernández-Valmayor, A. (2007). Universalizing Chasqui Repositories with a Flexible Importation / Exportation System. In Fernández-Manjón B., Sánchez-Pérez J.M., Gómez-Pulido J.A., Vega-Rodríguez M.A., Bravo-Rodríguez, J. (Eds.) *Computers and Education: E-learning - from theory to practice* (pp. 99-110). Berlin: Springer.
- (Sierra et al., 2007) Sierra, J.L., Fernández-Valmayor, A. & Fernández-Manjón, B. (Noviembre, 2007). How to prototype an educational modeling language. Actas del congreso *9th International Symposium on Computers in Education* (pp. 97-102).
- (Sierra et al., 2007b) Sierra, J.L., Moreno-Ger, P., Martínez-Ortiz, I., Fernández-Manjón, B. (2007). A Highly Modular and Extensible Architecture for an Integrated IMS based Authoring System: The <e Aula> Experience. *Software-Practice & Experience*, 37(4), 441-461.

- (Sierra et al., 2007c) Sierra, J.L., Fernández-Manjón, B. & Fernández-Valmayor, A. (Agosto, 2007). Language-driven Development of web-based Learning Applications. In: Leung H, Li F, Lau F, Li Q, (Eds.) *Advances in Web Based Learning – ICWL 2007, 6th International Conference on Web-Based Learning. 2007* (pp. 520-531). Berlin: Springer;2008.
- (Sierra et al., 2008) Sierra, J.L., Fernández-Valmayor, A. & Fernández-Manjón, B. (2008). From Documents to Applications Using Markup Languages. *IEEE Software*, 25(2), 68-76.
- (Sierra et al., 2008b) Sierra, J.L., Fernández-Manjón, B. & Fernández-Valmayor, A. (2008). A Language-Driven Approach for the Design of Interactive Applications. *Interacting with Computers*, 20(1), 112-127.
- (Slavin, 1995) Slavin, R. E. (1995). *Cooperative Learning: Theory, Research and Practice* (2nd Edition), Allyn & Bacon.
- (Sloep, 2004) Sloep, P.B. (2004). Reuse, Portability and Interoperability of Learning Content: Or Why an Educational Modelling Language. In R. McGreal (Ed.) *Online Education Using Learning Objects* (pp. 128-137). London: Routledge/Falmer.
- (Sloep et al., 2005) Sloep, P., Hummel, H. & Manderveld, J. (2005) The Learning Design Specification. In R. Koper, C. Tattersall (Eds.) *Learning Design - A Handbook on modeling and Delivering Networked Education and Training* (pp. 139-160). Heidelberg, Springer.
- (Specht & Burgos, 2007) Specht, M. & Burgos, D. (2007). Modeling Adaptive Educational Methods with IMS Learning Design. *Journal of Interactive Media in Education*, 2007/08.
- (Sprinkle, 2004) Sprinkle, J. (Febrero-Marzo, 2004). Model-integrated computing, *IEEE Potentials*, 23(1), 28-30.
doi: 10.1109/MP.2004.1266937
- (Stahl et al., 2006) Stahl, T., Voelter, M. & Czarnecki, K. (2006). *Model-Driven Software Development: Technology, Engineering, Management*. Wiley.
- (Stanchfield, 2009) Stanchfield S. (2009). *ANTXR: Easy XML Parsing, Based on The ANLR Parser Generator*. Disponible online en: <http://javadude.com/tools/antxr/index.html>.
- (Stayton, 2005) Stayton, B. (2005). *DocBook XSL: The Definitive Guide Third Edition*. SageHill Enterprises.
- (Steinber et al., 2003) Steinber, D., Budinsky, F., Paternostro, M., Merks, E. (2003). *EMF: Eclipse Modeling Framework*. Boston, MA, EEUU: Addison-Wesley.
- (Steinber et al., 2008) Steinber, D., Budinsky, F., Paternostro, M., Merks, E. (2008). *EMF: Eclipse Modeling Framework* (2nd Edition). Boston, MA, USA: Addison-Wesley.
- (Sterling & Saphiro, 1994) Sterling, L., Shapiro E. (1994). *The Art of Prolog* (2nd edition) Advanced Programming Techniques. Massachusetts, USA: MIT Press.

- (Sztipanovits & Karsai, 1997) Sztipanovits, J. & Karsai, G. (Abril, 1997). Model-integrated computing, *IEEE Computer*, 30(4), 110-111.
- (Thibault, 1998) Thibault, S. A. (1998). *Domain-Specific Languages: Conception, Implementation and Application*. Ph.D. Thesis. Institut de Formation Supérieure en Informatique et Communication. L'Université de Rennes 1.
- (Thibault et al. 1999) Thibault, S. A., Marlet, R. & Consel, C. (1999). Domain-Specific Languages: From Design to Implementation. Application to Video Device Drivers Generation. *IEEE Transactions on Software Engineering*, 25(3).
- (Thomas & Barry, 2003) Thomas, D. A. & Barry, B. M. (2003). Model-Driven Development: the Case for Domain-Oriented Programming. Actas del congreso *18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages and Applications, OOPSLA* (pp. 2-7).
- (Tidwell, 2008) Tidwell, D. (2008). *.XSLT, 2nd Edition*. O'Reilly Media.
- (Ulrich, 2007) Ulrich, W. (2007). *Architecture-Driven Modernization: Transforming the Enterprise*. Disponible online en: <http://www.omg.org/cgi-bin/doc?admtf/2007-12-01> (último acceso Noviembre de 2010).
- (Vantroys, 2003) Vantroys, T. (2003). *Du langage métier au langage technique, une plate-forme flexible d'exécution de scénarios pédagogiques*. Laboratoire TRIGONE - Équipe NOCE, Université des Sciences et Technologies de Lille: 182.
- (Vantroys & Peter, 2003) Vantroys, T. & Peter, Y. COW, a Flexible Platform for Enactment of Learning Scenarios. Actas del congreso *CRIWG 2003* (pp. 168-182). Springer Verlag.
- (Verbert & Duval, 2004) Verbert, K. & Duval, E. (2004). Towards a Global Component Architecture for Learning Objects: A Comparative Analysis of Learning Object Content Models. Actas del congreso *World Conference on Educational Multimedia, Hypermedia and Telecommunications*. Chesapeake, VA: AACE.
- (Vidal, J. C. et al., 2009) Vidal, J. C., Lama, M., Sanchez, E., Bugarin, A. & Novegil, A. (2009). OPENET LD: An Ontology-Based Petri Net Engine to Execute IMS LD Units of Learning. Actas del congreso *9th IEEE International Conference on Advanced Learning Technologies (ICALT'09)* (pp. 499-503).
- (Walmsley, 2007) Walmsley, P. (2007). *XQuery*. O'Reilly Media.
- (Walsh, 1998) Walsh, N. (1998). Entities: What are They Good For ?. *O'Reilly XML.com*. Disponible en: <http://www.xml.com/pub/a/98/08/xmlqna0.html> (último acceso: Febrero 2006).
- (Walsh & Muellner, 1999) Walsh, N. & Muellner, L. (1999). *DocBook: The Definitive Guide*. Sebastopol, CA, EEUU: O'Reilly.

- (Walsh, 2001) Walsh, N. (2001). Introduction to Docbook. Actas del congreso *Winwriters Online Help Conference*.
- (Walsh, 2004) Walsh, L. (2004). Using Extensible Markup Languages for the Single Source Delivery of Teaching Resources via Print and the Web: A Practical Example. Actas del congreso *Australian Society for Computers in Learning in Tertiary Education ASCILITE* (pp. 913-923).
- (Wang et al. 1997) Wang, D.C., Appel, A.W., Korn, J.L. & Serra, C.S. (Octubre, 1997). The Zephyr Abstract Syntax Description Language. Actas del congreso USENIX Conference on Domain-Specific Languages. Berkely, CA, EEUU.
- (Weitl et al., 2002) Weitl, F., Süß, C., Rudolf, K. & Burkhard, F. (2002). Presenting Complex e-Learning Content on the WEB: A Didactical Reference Model. Actas del congreso *E-Learn 2002 world conference on E-Learning in Corporate, Government, Healthcare, & Higher Education*.
- (Weller et. al., 2006) Weller, M, Little, A., McAndrew, P. & Woods, W. (2006). Learning Design, generic service descriptions and universal acid, *Educational Technology and Society*, 9(1), 138-145.
- (WfMC, 2002) *Workflow process definition language – xml process definition language version 1.0*. (Octubre, 2002). Workflow Management Coalition Specification WFMCTC-1025, Workflow Management Coalition (WfMC), Hampshire, UK.
- (W3C, 1999) W3C: World Wide Web Consortium (1999). *Web Content Accessibility Guidelines 1.0*. Disponible online en: <http://www.w3.org/TR/WCAG10/>.
- (W3C, 2001) W3C: World Wide Web Consortium (2001). *Extensible Stylesheet Language (XSL)*. Disponible online en: <http://www.w3.org/Style/TR/xsl/>.
- (Young & Halvorson, 2003) Young, M.J & Halvorson, M. (2003). Microsoft® Office System Inside Out -- 2003 Edition. Microsoft press.